

blenderart

MAGAZINE

Blender learning made easy

Animation Special !!

Tutorials

Spider rigging
Game character rigging
Animating falling feathers
Making of 'Age of Steam'
Making of 'Plumiferos'
Making of 'New Pengueon'
Creating a texture in Gimp

Articles

Project Orange special !!
Foss.in Report
'Stop Staring' - Reviewed

EDITOR

Gaurav Nawani
editor_blenderart@yahoo.co.in

MANAGING EDITOR

Sandra Gilbert
manager_blenderart@yahoo.com

WRITERS

Sandra Gilbert
Gaurav Nawani
Enrico Valenza
Juan J. Pena
David Lettier
Zsolt Stefan
Claudio Malefico Andaur

DESIGN & GRAPHICS

Gaurav Nawani

Website Design (underway)

Nam Pham

COPYRIGHT©

'Blenderart Magazine', 'blenderart' and blenderart logo are copyright Gaurav Nawani. 'Ask Blentuu' and 'blentuu logo' are copyright Sandra Gilbert.

All products and company names featured in the publication are trademark or registered trademark of their respective owners.

www.blenderart.org

Exclusive !!

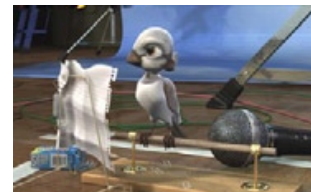
44

A look behind the 'Project Orange team'. In an exclusive interview with Sandra Gilbert.



34

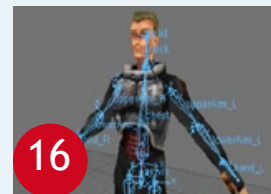
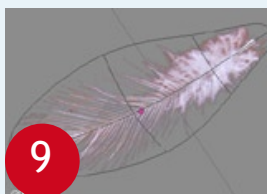
Behind the scenes of the 'New Pengueon' by Enrico Valenza



Plumiferos, an up-coming short from Studio Manos Digitale.

And more...

30



Learn how to animate feathers falling and how to rig a human and a spider



Sandra Gilbert

Managing editor

Welcome to the 2nd issue of Blenderart. This issue we are focusing on Animation. Animation is a actually a very large field, covering a wide variety of techniques and methods for adding motion, emotion and effects to an endless array of objects and characters.

The last few months has seen a major overhaul of the animation system, making animation much easier and less painful for even the least skilled of us. The ease of setting up a skeletal rig makes animating your characters less daunting, tempting even the most cautious of us to try our hand at animating.

We are very lucky in that Blender has so many tools for animating. There is a method and tool for just about anything you would want to do. For those of you who have not yet explored the animation capabilities of Blender, here are a few of the tools you may want to consider as you venture into the world of animation.

KeyFraming/IPO manipulation
Lattice deformation
Curves and paths
DupliFrames
Pose Mode/IK & FK Chains
Particles/Dupliverts (Especially with deflection and force options)
Soft Bodies

Rigid Body
Fluid Simulation
Shape keys

This is an impressive feature list and now you can use any of these alone or in various combinations to create your next animated masterpiece.

In this issue we are going to cover a few of these methods and techniques to get you up and animating. We are also bringing you up to speed on “Project Orange” and as a special treat, we even have a short interview with some of the Orange team.

On a personal note, the Blenderart staff would like to express our thanks to the entire Blender community for the overwhelming positive response and support of our first issue.

Happy blending!



Blentuu answers

*If you have a problem
or a tip you want to
share mail it to blentuu*

*manager@blenderart@
yahoo.com*

Q Hi there Blentuu,

I have a question for your section of the Blender art mag. I've used other 3d programs before like Maya, Max and XSI and they all have a tool that allows you to split a polygon in half. I can't figure out how to do this in Blender. I've tried using the knife tool to split a polygon but when it cuts, polygons on both sides of the one I tried to cut would have cuts in them too.

My question is; is there a way to cut a single polygon? If so, how? I hope my question makes it into the next issue. Thanks in advance,
- Alonso

A *Well the answer to that is no. In Blender, cutting a polygon will always create extra cuts/edges, but not all hope is lost. With the release of version 2.40, you get some control over the number of cuts with the new improved Knife tool. Or you can use the new Mesh Rip tool. The "Rip" tool can be used for tearing or ripping parts of a Mesh, in order to create seams to be filled in with new faces.*

Rules for the Rip tool are (taken straight from the release notes); Use short cut Vkey to activate it (a V resembles ripping :). It immediately enters Transform translation mode after.

It only works for quads, this because it internally uses the face loop-select to detect the proper faces that need to be ripped. If only 1 vertex is selected, the edge closest to the mouse cursor defines the direction of the rip seam and what part is being

grabbed. Note that in this case there are 4 possible rips, so positioning the mouse has to be done a bit careful.

When edges are selected, the rip seam is defined, so the mouse cursor position only defines which part is being grabbed.

With only 1 vertex selected, you can step by step rip a mesh seam; this also works on the edges of a mesh grid for example.

Q Hi,
I read and enjoyed your Blender art magazine, and have a question for your "Ask Blentuu" section.

I have an application in which I would like to use Blender, however, I have not been able to determine if the following feature is available. (For my purposes I would be driving Blender from a python script.)

Is it possible to have Blender calculate the volume of an object in order to calculate the object's mass?

Many thanks,
Charlie Taylor
charlie@sallyandcharlie.com

A *Python not being one of my greatest strengths, I did some searching around and discovered that there is a python script in progress right now that should cover what you need. It is named 'AreaVolume' and further information about its progress and usage can be found in the python forums at: <http://www.elysiun.com/forum/viewtopic.php?t=50510&highlight=areavolume>.*

Q **Hello Blentuu,**
I have a question I would like to see answered in the next issue of Blender art magazine.

I am new to Blender (using it about six months part time and have worked through the tutorials in the Blender 2.3 Guide twice now). I am just getting to the point where the application is familiar to me and can easily follow most well written tutorials. I come from a scientific background and I approach things the way I would, if I was executing a laboratory protocol (read procedure, gather materials and tools, follow instructions, collect data, analyse data, write a report). Now not having any formal art education I have been thinking about how to approach the work flow in Blender to produce a small animation project. Do the professionals have a general methodology that they follow: model characters, add texture and lighting, rig for animation, create animation, render animation files?

Some tips for a noob regarding typical work flow that professionals use when approaching a small project would be of interest to me. I like the first issue and am happily working on the included tutorials. Thanks for the magazine hope to see something on work flow in one of your upcoming issues.

A - Matt

Any work flow you use will be of course tailored to your individual work style. That being said, a commonly used work flow.

A) Planning Stage.

B) Story line.

C) Concept sketches, perhaps even do up clay models.

D) Story boards (even if only using stick figures).

E) The next part is done in progressive stages of detail. Set up your shot/scenes with basic primitives, to get an idea of spacing and size Do a quick render with basic colouring and lighting to judge timing.

F) Model low poly versions of everything, refining colouring, lighting timing issues. Add in rough dialogue and sound effects. Add progressive amounts of detail, materials and texturing and lighting, fine tune the timing and sound issues

G) Final render

This type of setup works best in a team environment, with everyone assigned specific tasks, but can be modified to an individual effort.

-- **Blentuu** ■

Blender 2.40 released



The Blender version 2.40 has been released with an overwhelming list of new features that will keep all of us busy experimenting and learning. Although there were a large number of new features, most of the attention has been given to the Animation code improvements and rewrites.

With such features as a new IK System, Stride Support, Shape Keys, IPO Drivers, Skin Weighting and Rigging improvements, animation has become not only fun, but a lot less painful. The new Particle Guides and Shaded Particles have added a whole new dimension to hair/fur/grass.

Having already played with a lot of the new features, I can tell you that it is well worth the download. A few features have been moved, but not removed. So before you panic, look in the new Modifier Stack, in the Edit Buttons. Most all effect features such as

SubSurf, Decimate, Wave, Build etc have found a new home there, as well as a few new ones. There have also been rather exciting changes and improvements to the UV/Image Editing tool set. A new 'Live LSCM' gives instant feedback while UV mapping, while a number of new tools allows cloning and smudging of images within Blender itself. TIFF support has also been added for import and export.

News of this release has already reached all corners of the Internet, even appearing on the front page of the prestigious CGTalk website, adding to the already growing popularity of Blender. So, if you somehow managed to miss the news of the release, head over to www.blender.org and grab yourself a copy. Don't forget to read the release notes as there is a wealth of information contained in those pages and you wouldn't want to miss out on even one of the new features.

Every Blender release is followed by a 'bugfix' release. From the developers meeting notes, it looks like 2.41 may see the daylight next week: Instead of "2.40a" we drop the a/b convention and the 2.40 bugfix release will be called 2.41 ■

Gimp in news

The Gimp team has released the latest bug fix release of Gimp as the new stable release in Gimp 2.2.10 and also released Gimp 2.36, the beta release for the upcoming Gimp 2.4.

Although the Gimp stable release doesn't add anything in new features, it is recommended to upgrade your older Gimp 2.2.9 or 2.2.8 versions with this.

For those seeking to play with the upcoming features in Gimp you can also download the Gimp 2.36 in source form from the www.gimp.org website.

There is also a windows version of gimp 2.35 available at source forge, though it is termed as unstable it is pretty useable and lets you experience the all new organised menu structure, lots of elementary goodies that finally made it into the text tools and much more, go play with it and provide feedback to the Gimp team ■

Foss.in report

The Linux Bangalore has always been a big event in India. Bangalore being the IT capital of India, has tremendous interest in FOSS. This year the LB was aptly renamed and transformed to FOSS.in as it has grown greatly over the years and due to the amount of attendance it commands it is no longer correct to call it a local LUG event any more.

Day 1

I was there as a Blender user presenting the ever increasing quality and features of Blender at a booth. The event was launched by Atul Chitnis, one of the biggest supporters of FOSS in India.

After attending the inspirational talk by Alan Cox, we went to fetch the machines which we were going to use for our demos or projects. We got brand new machines from the warehouse. Then I realised that I had a problem on my hands, this being the first of any such event for me, I didn't bring a copy of Blender with me! Although I had duly burned all necessary stuff for demonstration a day before. No worries, just next to me was the strong contingent from Delhi LUG and they had everything I needed ;).

After a short while I started demos of Blender animations, using the recent Suzanne award winners. Slowly many people gathered and started to discuss among themselves as to how Blender stands amongst other commercial counterparts.

I did not answer any of those queries, but promptly

switched the animated demos to the rendered artwork. Their queries immediately changed to: how was this stuff done in Blender? How much time it would take to do so? Exactly the questions I was waiting to answer :)

Things were going fine for us, though we did not have net access until Atul came in and suggested that we move to the 'Business Expo booth', as there was plenty of empty space and ready net access there. By the time lunch ended we had shifted there. At the end of the day the crowd waned.

Day 2

I was prompt to present myself at the event the next morning, however it seems morning does not wake people faster. Since I had nothing to do, I went and attended my first FOSS event of the day Ahem... 'LinuxChix'. After attending the event I returned back to my stall and was told that a few guys were interested in Blender and they were waiting for me for some time. After a while a small group of four guys came over, and asked what Blender could do. I showed them some stuff, Enrico's 'Penguin Dance' was a favourite and so was the 'Cycles' animation.

Shortly after noon a group of youngsters presented themselves at the stall. They were the enthusiastic type and one of them wanted to actually see how it was done. There began my first live Blender session and boy I was not aware of the usefulness it had.

Day 3

This was the busiest day for me. I gave about 5-6 Live Blender sessions, it was as if my stall had a high PR rating through 'word of mouth'. Sartaj Kang, one of the KDE developers (and few others who have been watching the small groupings with amusement) were of the view that I should have been giving a full



talk instead, at other times they even wanted a BOF entirely for themselves. I nodded in agreement but luckily they were just too busy to attend one ;)

Just next to us was an HP stall. There they were demoing a Blender version running on a modified xserver which made use of three monitors and gave a 'large desktop real-estate' for Blender. There was Mr. Manjunath from HP, with whom I discussed various issues under which HP could be helping OSS. But as they always say a commercial company has commercial interests in mind, well rightly so, but at-least he agreed in theory that there can be a chance that HP can make use of Blender's growing capabilities. The talk went on to comparing the powerful features of other commercial apps with Blender, which came back to the development of features needed and when I told him about the speed of actual development, I could see him getting excited.

Day 4

Again a slow start in the morning, but this was one of the most productive days of all. A professor from one of the Goa Universities wanted to learn the basics of using Blender himself so that he can introduce this useful tool to his students. There was only one problem though, he was looking for CAD features which Blender does not readily offer. I discussed a work around with him, like importing the mechanical drawings through images and building models with reference. They might not be accurate, but 3d models will keep student's interest. I also suggested to him other OSS CAD tools. Despite that, I spent a good 1-2 hrs. detailing for him how to use Blender, and I must say he was really interested in learning. I hope he can make use of the small learning session.

Then there was Vaseem a engineering student, who was interested in learning whether Blender can be used for his academic project on 'photo geometry'. We discussed a lot on the issue and finally he agreed that using python with Blender might help him.



Finally I met an very interesting person: Shree Kumar. He had worked on Blender scripting and was the one those guys who made those Blender demos running on the HP Cluster (or was it some thing else ?) And if I recall right, was one of the people who customised the Xserver to use multiple monitors asynchronously on that machine. More importantly, we developed an mutual understanding that I can misuse his talents to correct errors on python scripts which fail on my machine ;)

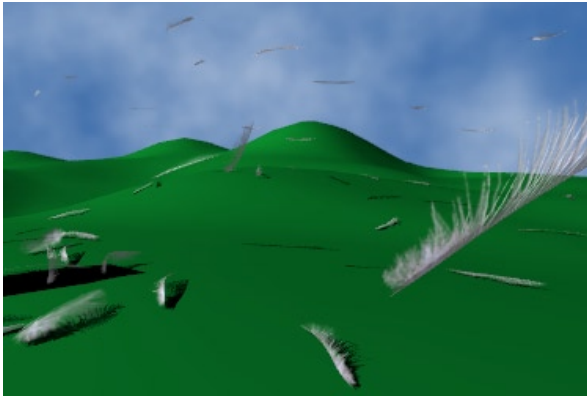
At the end of it, I was filled with vigour, that I will return next year with a much larger role than this.

If Blender could generate such a good amount of interest despite being announcing as an entry very late in the event schedule, then it surely will have a much better presence with a full fledged talk on the subject, and more importantly a developer friendly BOF on which I am sure Tom (Letterrip) will be nodding his head in agreement ;)

-- Gaurav Nawani ■

Falling Feathers Tutorial

-- Juan J. Peña/Apollux



Introduction

In this tutorial you will learn how to create the effect of very light weight objects freely falling to the ground. Such light objects could be feathers, rose petals, confetti, etc. We will use the feathers as an example because they are harder to animate, as well as they make for the most complete example.

This tutorial is intended for the intermediate and advanced users. You are expected to already have a working knowledge about IPO Curves, Dupliverts and Particles. Blender 2.37 or later is required.

The Desired Goal

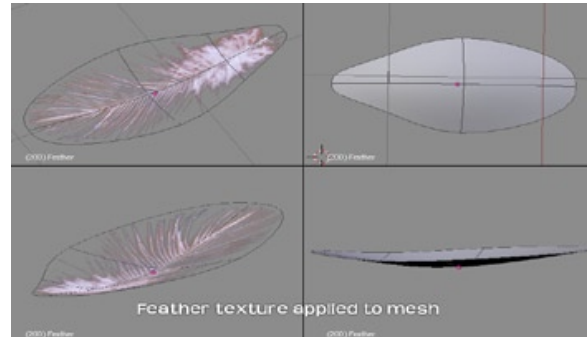
During an especially dramatic moment in the plot, the main character is walking quietly across a green

field. The sun shines, wind blows slowly, the music is melodramatic, and everything looks like a dream... suddenly the main character sees a white feather falling from the sky. She catches it with her open palm, looks up and sees thousands of white feathers coming from the sky. She smiles, somehow now she knows that the conflict is over and her beloved one is fine. The above scene should be familiar for many readers... with some small variations it has been used in many movies, games, novels, etc. This tutorial will teach you how to deal with the feathers... the character itself is outside of this tutorial's scope.

What we need

Obviously, the first thing we need is a CG feather. I quickly made mine by taking a mesh plane, cutting it a few times and merging the tips. I got something like this:

Notice that it has depth (so that when you see it from the side you don't see a depth-less plane) and (pay attention to this) the object center is placed below, not at the usual center, of the feather.



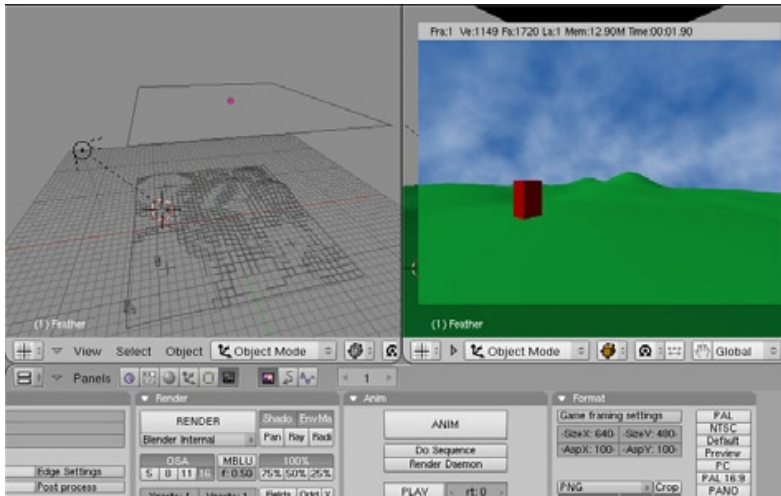
Giving it a nice white material would be cool, and if you feel like texturing it with an alpha enabled feather photograph, that wouldn't hurt either. But those are cosmetics and won't affect the animation results, just enhance it. (However, If you do go for the texture, remember to set it to affect the "Col", "Nor", "Ref", "Spec" and "Alpha" channels in order to get a convincing look and not just a feather painted on a piece of glass.)

We also need a CG environment where the whole sequence takes place.

For that I took a mesh plane, subdivided it a few times and raised some points to add to the terrain illusion. Drop in a green material, proper lighting and a sky texture to the world settings.

Of course, we would like to have an actual character interacting with the falling feathers (that's the point of the whole scene, isn't it?); but that would mean getting also into modelling and character animation, which are outside this tutorial's scope. Instead of our lovely lady we'll use a red cube as our stand-in actor.

With all that set we are finally good to go! By now you should have something like this:



Since this is an animation tutorial, I'll grant you some leverage with the modelling. You can get a ready to use scene at the blenderart website. If you don't feel like modelling all that stuff right now.

The First Instinct

Most likely your first instinct would be to create a particle system, set the ground and our stand-in character as particle deflectors, and let Blender do its magic. Let's explore that road.

We need to create a particle emitter that would cast the feathers; a mesh grid should do nicely for that. Place the emitter grid outside of the camera's view and place the feather's object center in contact with the grid's object center.

How-to?

Select the emitter grid (Right click on it)

Object | Snap | Cursor | Selection

Select the feather (right click on it)

Object | Snap | Selection | Cursor

Now make the grid the parent of the feather, activate Dupliverts for the grid and add a Particle system to the grid. Remember to set some negative Z force so that all feathers eventually fall into the ground.

First thing you will notice is that unless you set the "Key" value to a high number the feather's fall will be erratic to say the least. The best

results are with "key" set to the maximum value of 100, but you can get usable results setting it around 80 or above.

Second thing you should notice is that even with the deflectors damping set to the maximum, feathers still slide over the ground, just like if the ground was slippery ice, eventually most of the feathers find their way to the edge and fall into the void... not good!

The last, but certainly not the least important, is

that collision detection for particles isn't extremely accurate and you get plenty of feathers going briefly through the ground and then resurfacing. That may be OK for feathers further away from camera, but it would totally ruin any close up take.

Not everything is bad with this approach. The particle system allows us to introduce a certain amount of randomness to make the whole thing more believable. Specifically, by making the particle motion path B-Spline interpolated and giving each feather a random starting speed the illusion of rain increases a lot. The feathers still behave kind of fake, but we will work on that shortly. Special care should be taken with the Vect" setting, as after certain point it tends to introduce a 90 degrees turns in the particles/feathers.

Summing it up, these are the settings I am talking about:



We will seek a way to overcome the drawbacks of the Particles approach, while retaining all the good stuff.

A Better Approach

As mentioned before, the particles approach for the task at hand has some serious drawbacks, the collision detection being the worst of them. However, particles aren't the only objects with Collision detection within Blender. We also have collision detection for Soft bodies and Hard body dynamics inside the Game Engine. I find the collision detection implemented for Soft bodies far more refined, so we'll use that.

First thing that you should be aware of is that there are two basic modes of using the Soft bodies feature. You have the "Goal" and the "Spring" methods.

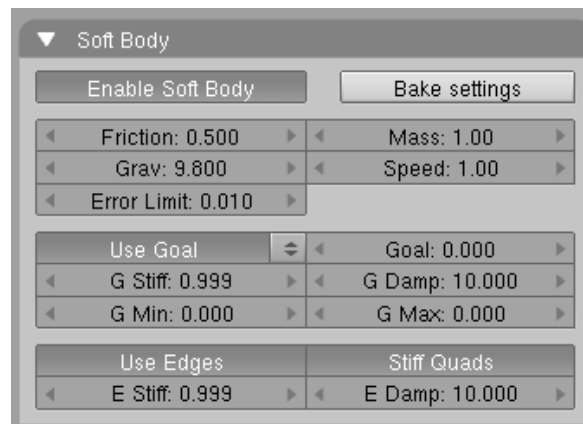
Without entering on technical details, the "Goal" method allows the objects to be animated using the standard ways, thus giving you almost total control over the animation result. With the "Spring" method most of the results comes from the dynamics system and you have somewhat less control over the final results (although they usually look more realistic than what you could get doing the animation by hand). Of course we want a good compromise between the two methods.

Ideally, we want to be able to influence the fall by introducing local transformations while the feathers are moving, but we also want that the system to take care of all the collision detections and any other dynamic forces, like wind gusts, that we might put in the scene to enhance it.

So this is what I call a good compromise: (Image on right)

Please notice these settings:

1. We had set the gravitational acceleration to 9.8 (real gravity), so we are telling the Soft body System that it should take care of the downward motion of



the feathers.

2. It is indeed a "Goal" system, thus we can pass local transformations to it (extremely important as you'll see), but...

3. It has both "Goal Influence" and "Goal Stiffness" set to zero, so the actual mesh deformations will come entirely from the dynamic system.

4. "Goal Dampening" has been set to the maximum value to help achieve point #2.

5. To prevent the feather mesh from deforming in unpleasant ways we are using both "Edges" and "Stiff Quads", with its settings all set to the maximum pos-

sible values. Likewise, "G Min" and "G Max" are both set to zero, just to be sure.

It goes without saying that the Soft body system should be applied to the Feather itself, not to the emitter grid. What does need to be said is that until some technical issues are solved, Blender "cannot render duplivered particles with softbodies".. So, for now, disengage both the particle system and the Dupliveres on the grid. The easiest way to do it is by un-parenting the feather from the grid and hiding the grid on an unused layer. Don't erase the grid or its particle system because we will need them later on.

It also goes without saying that both the ground and the actor should become soft body deflectors so that Blender can calculate the feather collision with them.

Starting from Blender 2.40, soft body collision calculations are no longer limited to mesh objects. Now collisions can be calculated between two or more soft body objects. In other words, no more feathers going through it's neighbour's mesh.

Keep in mind that adding those extra collision checks increases by a great length the time needed for calculations, so I decided not to do it now. If after you have rendered your first animation test you realise that you have no other choice than go ahead with it, but you are warned that it will take a long while to finish each frame.

Likewise, to cut down the calculation time I temporarily neglected to turn the ground into a soft body deflector, at least until all the extra magic is done. Just remember to turn soft body deflection back on; otherwise the feathers will pass right through the floor.

Refining It Up

Now that we have solved the main issue of the collision detection it is time to start working on the fine details.

Have you ever watched a free falling feather, or a light weight leaf coming off from a tree branch? Then you know they don't fall in a straight line, they have some sort of soft rocking movement while they spin over their gravity center.

Certainly we could animate those movements by hand using motion paths and rotation key frames, and it would be easy to do for just a couple of feathers. But we are talking about hundreds, maybe even thousands of free falling feathers, thus we need a more flexible and less work intensive way to do it.

Delta IPO's

Highly underrated, but extremely useful those IPO channels are. "Delta" is a letter from the Greek alphabet and is commonly used in math and physics to represent how much a given magnitude varies over a given time. Translating it to common language, it measures how much something changes from its initial state.

Blender offers delta IPO channels for location, rotation and size, in all three axis. Select any Object, open an IPO window and look for the channels whose name starts with "d".

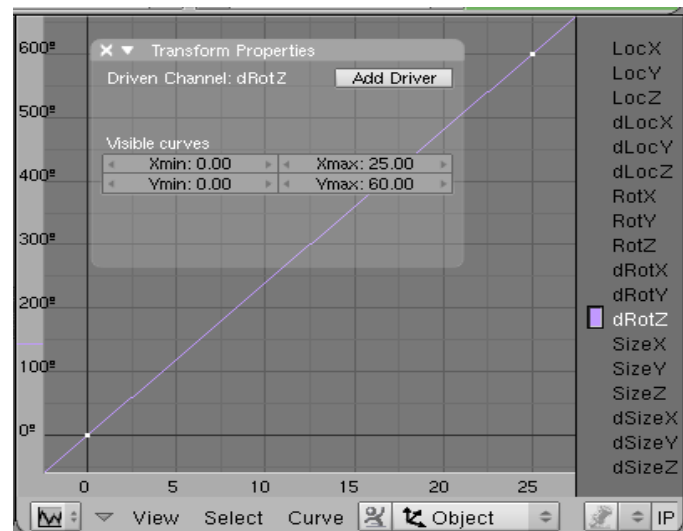
<-screen shot with the delta channels.



I'll be assuming that the reader already knows how to edit and add key frames to an IPO curve straight from the IPO window. If you need a reminder then I

strongly suggest you to go and re-read those chapters on the official documentation.

So now, with the feather selected, pick up the "dRot Z" channel and draw a curve for the spinning movement. Don't over do it. I have chosen to draw a simple 2 point curve, one point is at (0,0) and the other is at (25,60). This will cause the feather to rotate 60 degrees each 25 frames. With the curve set to Cyclic Extrapolation the feather will keep spinning forever. (Screen shot with the dRot Z curve)



Before we get any further, probably you are asking why use “dRot” instead of “Rot”? Quite simple, had we used “Rot” instead, it would cause every feather to rotate on exactly the same way... all feathers, at all times, would have the same rotation! It might look good on a choreographed dance sequence, but that is precisely what we want to avoid on something as random as a rain.

Using “dRot” means that whatever rotation the IPO dictates is added to whatever rotation the feather previously had. So, if we give to each feather a different starting rotation, its rotation will keep being unique through the entire sequence, which is a more desirable result.

Continuing on with the refinements, now that we have the feather spinning over its own gravity center we also need to introduce the soft rocking movement that is so signature to this kind of object. For that the “dLoc X” and “dLoc Y” will be of much help. Draw some IPO curves like the ones on the screen shot, making sure that you cross the zero line a couple of times. When you are done set the curve to Cyclic Extrapolation, so that the rocking movement will keep going on forever.

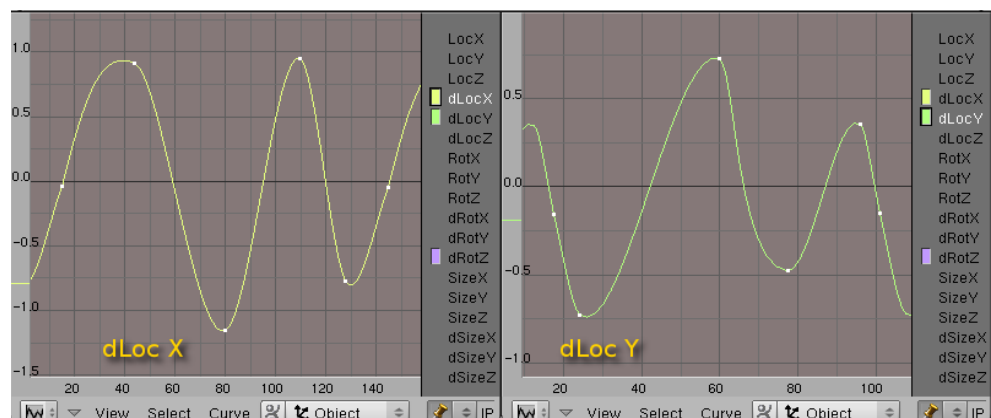
For both curves I tried to remain within +/- 1 Blender unit around the zero line. Anything more than that and it will start looking unnatural. Of course, the actual values also depend on the weather conditions you are trying to represent and the global scale of the scene. You should also double check that when you zoom out the curve it stays centred around the zero line.

It is quite easy to draw curves that look OK when you are working on them, but that tend to progressively deviate from the zero line. After some testing and trial you should be able to get the hang of it, it is not that difficult, trust me.

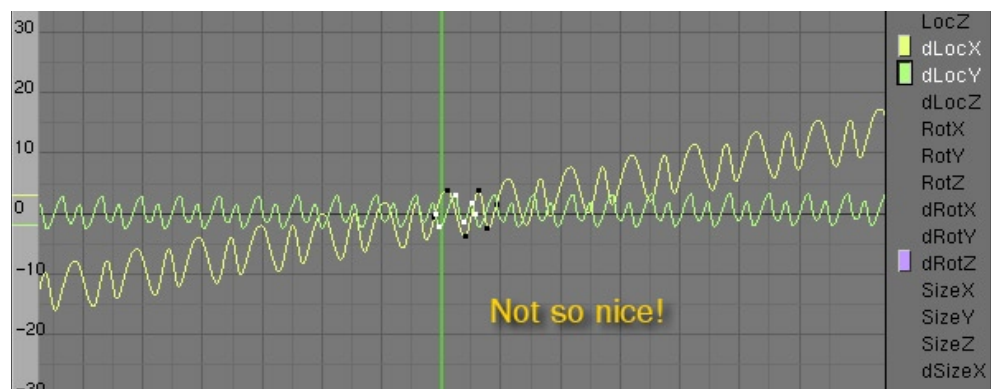
Here is an example of what you should avoid: (Image bottom right)

In the screen shot you can see that the dLoc X curve progressively drifts away from the centre, so that over time it gets harder to predict where the feather is going

to move next. The situation can be rectified with the techniques we’ll see on the next page, but better avoid the problem if you can.



Screen shot with the dLoc curves-->



Further Refinements

If you playback the animation one frame at a time you'll see that now we have a much more natural motion for the feather... now it is time to start polishing the final details.

First detail is that right now we only have one single feather. Do you remember the emitting grid and its particle system? Yes, the one we put away on an unused layer some pages back. Now it is time to bring it back. As before, parent the feather to the grid and let the grid create as many duplicates as you need. Don't be afraid to play with the settings. The goal now is to create as much randomness as possible.

If you render the scene as it is right now you will see that all the feathers are missing on the render. Remember what I told you about Blender having problems with dupliverter soft bodies? Well, this is one of those cases.

To work around the issue we need to make all the dupliverter feathers real duplicates. Select the grid and press [Ctrl]+[Shift]+[A] or use Object >> Clear/Apply >> Make Duplicates Real

You should expect a major slow down right after you make the duplicates real, because now the Soft body system is calculating hundreds or thousands of feathers at the same time. Playback the animation. Is everything fine? No? The

duplicated feathers keep falling on a straight line? Don't panic !!

When you make the duplicates real Blender doesn't copy over the animation IPO's, just the meshes and materials. We will solve it by selecting all the feathers and making sure that the last one we select is the original feather (so it is the active object of the group). Then use Object >> Make Links >> Object IPO. If you playback the animation or check each feather on the IPO window you'll see that everything is fine now.

It is time to do a test render. Since the IPO curves are just one set linked to all the feathers, fixing any mistake on the IPO channels is as simple as selecting one of the feathers, doing the needed changes and all the changes will pass over to all the others.

Note: Depending on what you want to achieve, on occasions it may be worthy overriding the soft body system just before making the duplicates real... For that just go back one frame before using the "Make Real" command. I don't advise to actually click on "Disable Soft bodies" before making them real because then you would need to manually select and re-enable it for each feather. When we are talking of hundreds of feathers that isn't a trivial task.

The Final Touches

What about adding a few gusts of soft wind and maybe a force field vortex around the main character

to make the scene even more natural and emotive? Don't forget to place some of the air currents near the ground in order to justify why the feathers are still spinning and moving around even after they touched the ground.

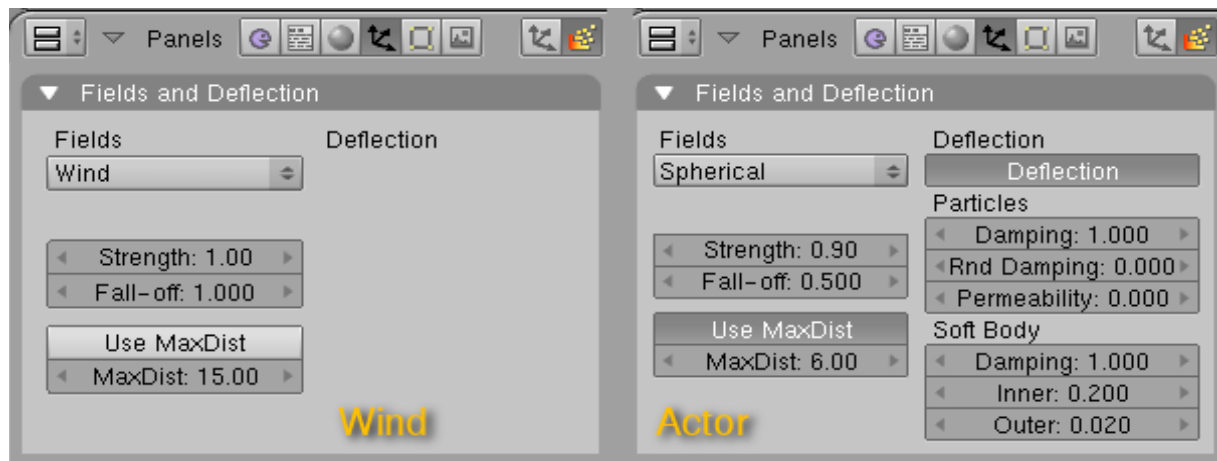
For this particular case, these are the wind and main actor's force field settings that are giving me the best results, but that is more of a personal preference than anything else.

If you previously disabled the Soft bodies' deflection for the ground and/or actor, remember to re-enable it before doing your final render. On my experience, rendering with deflection enabled for the ground increases a lot the time needed to render each frame. On my machine, the change of speed is really dramatic, jumping from 17 seconds per frame to almost 4 minutes per frame!

Photo Credits:

The feather textured packed within the .blend is based on a royalty-free feather photograph that I found at "Image*After" (<http://www.imageafter.com/>).

Other places where you could find royalty free feathers pictures are "Stock.Xchng" (<http://www.sxc.hu/>), "morgueFile" (<http://morguefile.com/>) and "gettyImages" (<http://creative.gettyimages.com/source/home/home.aspx>).



If you have any doubts about how something is done, the finished .blend is available at (blend file included with this issue of Blenderart) for you to inspect it. This is how we reached the end of our brief rain-making tutorial. For any comments, don't hesitate contacting me at apollux@apollux-designs.com, or look out for me at the various Blender forums on the net.

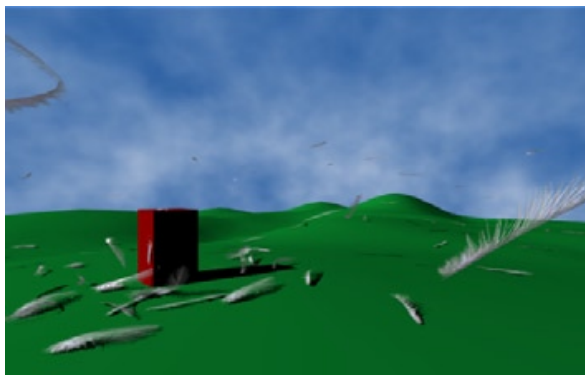
Ooh... By the way, do you remember the single feather that lands on the main character's hand at the introduction of this tutorial? That is done with a normal path curve. Select one of the feathers and parent it to a path leading to the open palm ■

Conclusion

So far we have only scratched the surface of what could be done with this technique, and that was the purpose of this tutorial. Now it is up to you to keep refining, customising and improving the idea. These are just some pointers of what could be done to further enhance the scene:

1. An animated sky with moving clouds and changing hues (make it match the direction of your wind force field).
2. Better grass for the green field.
3. Increasing the feathers count.
4. Adding more variation to the feathers, right now it is the same feather being copied over and over.
5. Breaking up the feathers cloud into smaller groups, each one with its own set of IPO's.
6. Replacing the red cube with an actual animated character.

After adding some wind gusts and playing a little with the sky, this is what I got as my final animation (movie location to be announced). It took nothing less than 14 hours to render its 300 frames, so I hope you like it.



Juan J. Peña M. (aka apollux) is a 27 year old Architecture student in the Dominican Republic. Being a long time Blender user has led him to become the Blender Linux FAQ maintainer. In his spare time he follows his life long love of amateur photography. Before he re-entered University to study architecture he was a former Computer Science (for Kids) teacher.

<http://apollux-designs.com>

Character Rigging and Animating

-- David "IPKnightly" Lettier

Introduction

Providing movement to your game art assets is an absolute must to providing a depth of immersion. Some items such as rocks, vegetation, buildings, etc can and should be static, however, for your main character and surrounding NPCs (among other game art assets) need to be able to move in a realistic way to trick the player's mind even further into believing the virtual world you're building.

To help in moving your characters, you should be familiar with the terminology or "jargon" that comes with it. Animation, whether 2D or 3D, is creating a sequence of frames or pictures that are subtle changes as you progress in the frame number or count. By displaying these pictures in rapid succession you create the illusion of animation or movement. KeyFrames are single frames of animation that discretely define a position or point(s) in space to interpolate from (read below). In other words, they signify a subtle to major change in movement in the desired animation. For example, you're creating a bouncing ball animation. Thus, you create three KeyFrames: one when it is in the air, one when it hits the ground, and one in the air again but slightly lower than before. However, this does not define the frames of animation in between; to help with this interpolation is needed. Interpolation, is "a method of constructing new data points

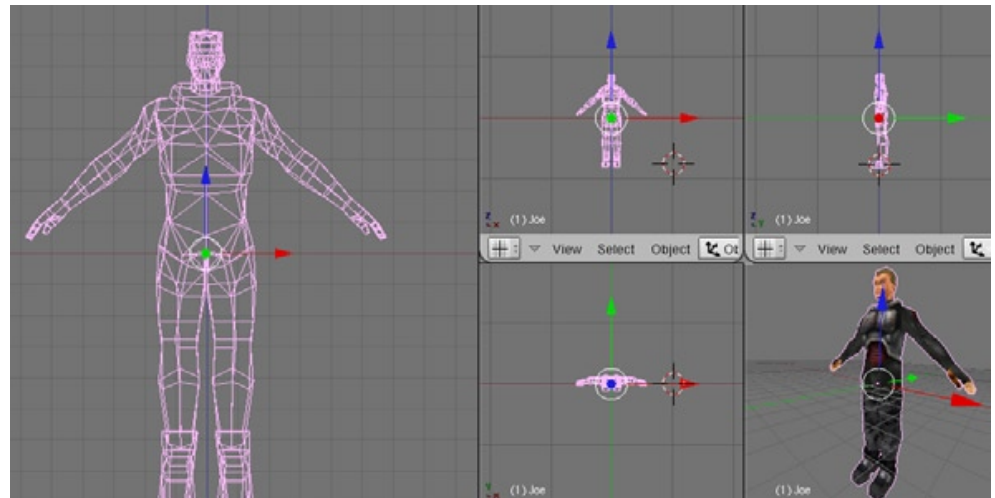
from a discrete set of known data points." As in the ball example, it defines the points/positions/frames the ball needs to be in based on the starting and ending KeyFrames or positions that are discretely or distinctly defined. By using interpolation, you do not need to define/create every frame of animation as say in 2D animation. Per-vertex (or morph-target) animation is used to provide movement to character meshes and the like.

By moving the individual vertices of the mesh each KeyFrame, realistic movement can be achieved by interpolating each vertex between KeyFrames. Skeletal animation uses a bone or armature system with each bone assigned to a vertex group. This process of assigning bones to vertex groups is called "skinning" as you "skin" the bones with the mesh. Moving these bones, moves their assigned vertices or vertex group. Defining bone positions each KeyFrame, the bones and thus vertex groups are interpolated between KeyFrames.

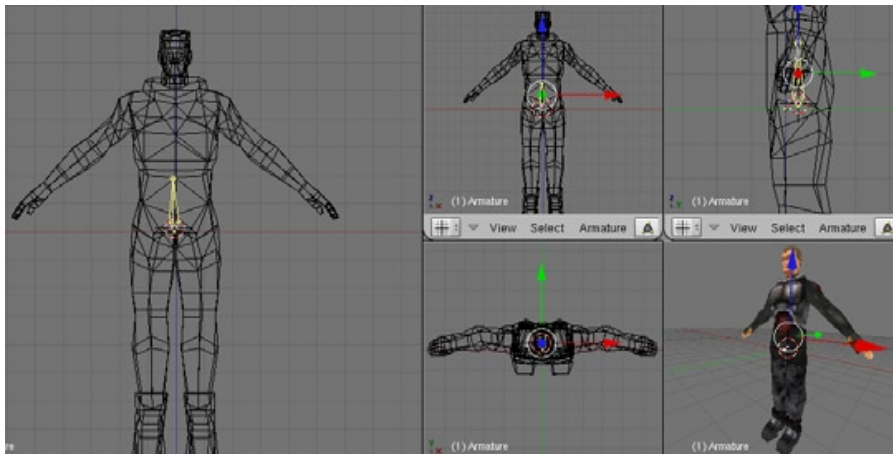
A major downside to skeletal animation is that you lose the fine control over where each vertex is placed each frame. Also, the mesh does not always skin correctly as in a bulg-

ing bicep if you moved the forearm bone for example. On the other hand, per-vertex animation is very time consuming as you must define each vertex's position, however, you do have complete control. For the following tutorial we will be using skeletal animation.

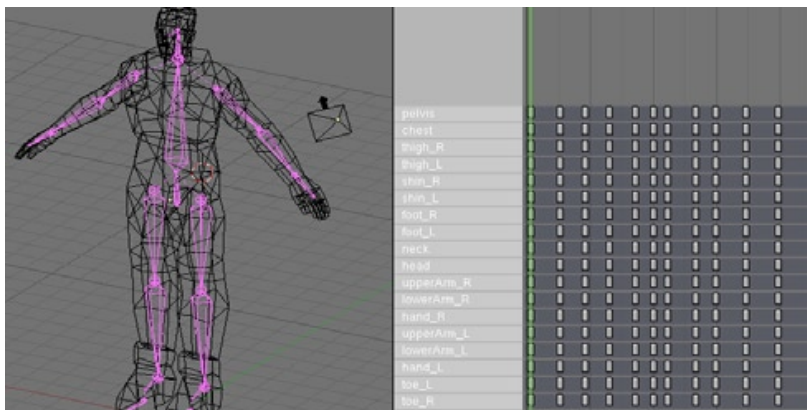
Begin by opening Blender, press [A] then [X], and then click OK? >> Erase selected to delete the default cube, light, and camera. [NumPad-1] to switch to the front view. [Shift-F1] to append to your current Blender file (.blend). Find the directory you placed the provided .blend file. Click on the .blend file and look for the Object section. Click Object, then Joe, and then Load Library. Your screen should look similar to this. See Image below.



Next, make sure the cursor is centred on the pelvic area of “Joe” from the front and side views. [Spacebar] Add >> *Armature*. This is going to create a new armature or bone system for our mesh “Joe.” You should have something very similar to the image.



Now we will begin to build a skeletal system very similar to the image below.



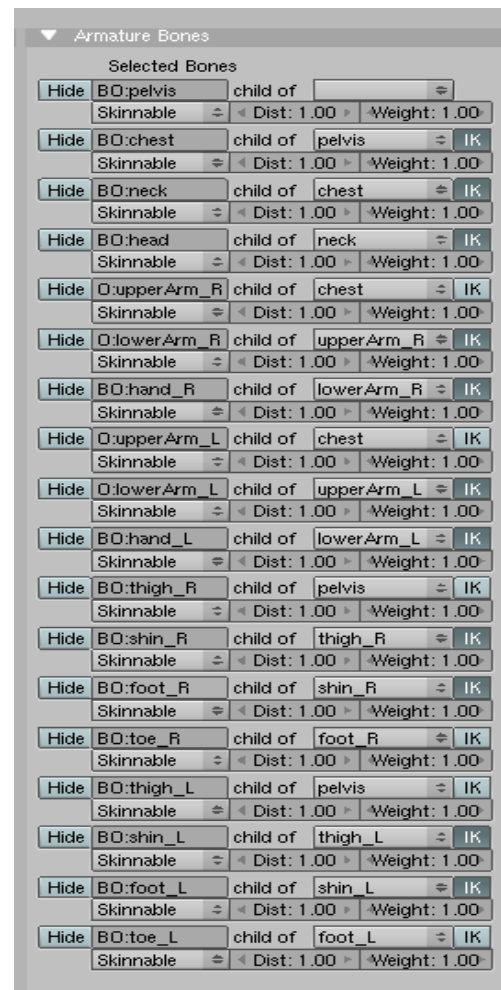
To do this, extrude [E] OK? >> Extrude bone segments the bones (beginning with the pelvic bone) to create a chest bone, neck bone, and head bone. Now place the cursor at the shoulder areas and [Spacebar] Add >> *Bone* and begin extruding the bones to create an

upper-arm bone, lower-arm bone, and hand bone. Duplicate this for each arm (two in total). Remember to follow the picture for reference. Once you have finished with that, place the cursor at the top of the thigh area on the mesh and create a thigh bone, shin bone, foot bone just like you did for the arms. Now place the cursor (viewing from the side view [Num-

Pad-3]) and add a new bone for the toe bone. Make sure to add one for each foot.

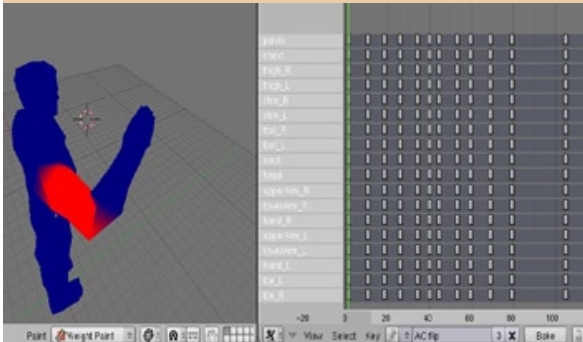
OK, now that the basic layout for the skeletal system is complete it is time to parent certain bones to others and name each bone. While still in edit mode for the bones press [F9] to get to the edit mode buttons (if you were not already there). Look for Armature Bones >> Selected Bones in the Buttons Window. Make sure to select all bones [A]. To make it easier to explain, match

your bone names and parents to the image below. Note, that a typical naming convention is to add “_R” or “_L” to either right or left bones respectively.



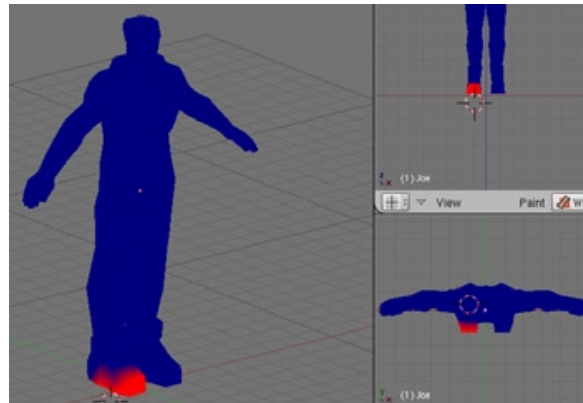
To recap we have appended our mesh object, created a basic layout for a humanoid armature system, and named/parented the bones together. Next we will assign the vertex groups to the armature system via “weight panting.” (see box)

Weight painting: Weight painting in Blender is the process of painting of vertices which get assigned to the different bones along with their “weight” value or rather the influence level that the bone has on the vertex; blue being no influence (weight 0.0) while red being full influence (weight 1.0). See Image below.



Trial and error is going to be the technique of choice until you become familiar with the mesh and weight painting in general. Begin by selecting the mesh [RM] (right mouse button) then select the armature you created with [Shift] [RM]. The mesh should be a dark pink while the armature is a lighter pink colour. Now with both selected, press [Ctrl-P] *Make Parent to >> Armature* then *Create Vertex Groups? >> Create From Closest Bones*. With the bones parented to the mesh,

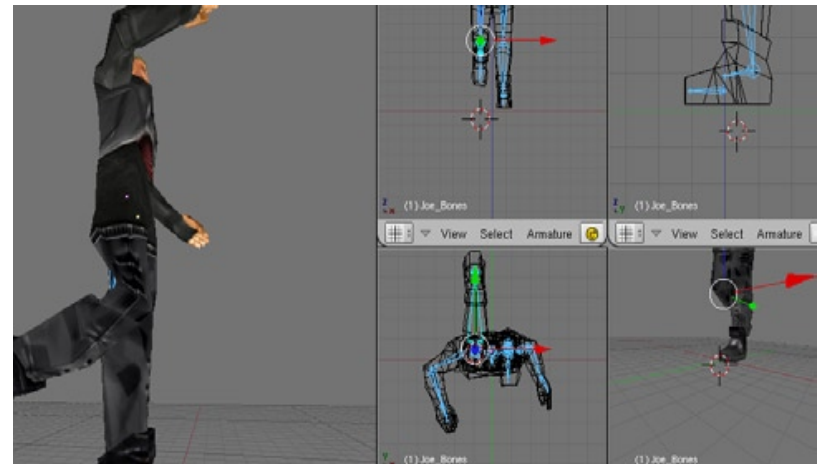
select the mesh only and enter weight paint mode. You should see something similar to this



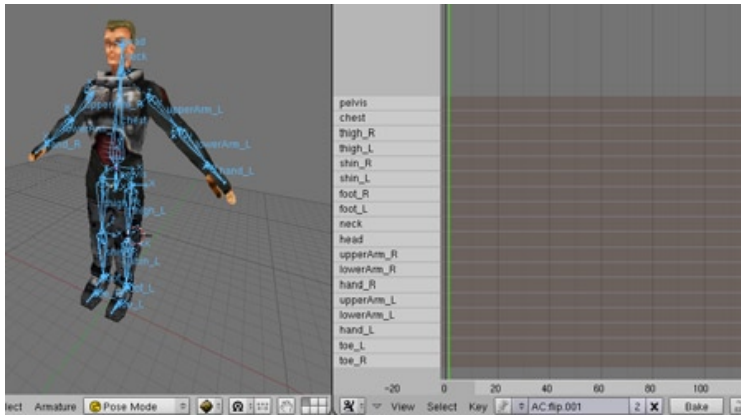
In rare instances, you will not have to change the selected vertices that were chosen when you selected *Create From Closest Bones*. In most cases you will have to at least select or remove some vertices from the vertex group. Other times, it is best to erase all selected bones by painting over the selected vertices with a weight of zero and full opacity. As said before, weight painting can be a bit of an art. Choosing which vertices belong and which do not can be difficult at times. Select the bone you would like to begin weight painting under *Links and Materials >> Vertex Groups* in the Buttons Window view. To paint the influence or weight value, [LMB] over the

mesh in the 3D View. As you paint over your mesh, you will notice the vertices being coloured either red, green, blue, or a colour thereof based on the Weight value under Paint in the Buttons Window view. These colours indicate the influence the bone has on the vertex. For most applications, painting in red seems to be sufficient enough—you may find otherwise. Lastly, concerning weight painting (or vertex groups in general), you may come across what is known as “pinching” or where too many vertices are attached to too many bones creating too much deformation. See below.

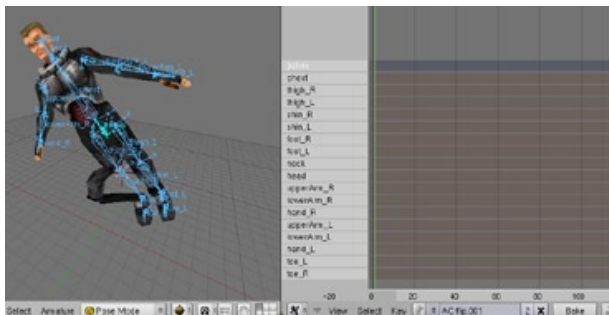
To correct this, remove the vertices that should not move along with the bone you are rotating. Again, this will be mostly trial and error.



We finally arrive to the “fun” part—pose mode or the actual animating. Select the armature with the [RMB] and enter what is known as Pose Mode under the different modes such as Edit Mode or Object Mode. Your screen should look similar to this



.Before you begin, select the root bone or pelvis bone [RMB] and rotate with [R]. This will rotate the entire mesh/armature. If you see any vertices left behind you must select the mesh and enter weight paint mode assigning the unassigned vertexes to their appropriate vertex groups. See image below.



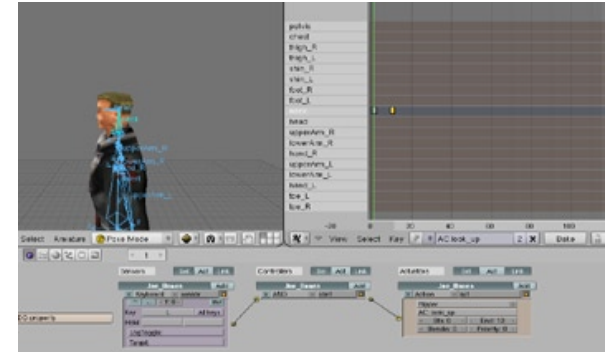
If you wish to ever return the armature and subsequently the mesh to original position press [Alt-R] to clear rotations applied while in pose mode. Note; that you can also select Rest Pos to return to the original position, however, this will not allow you to apply any transformations and is not permanent.

When animating, it is very helpful to stand in front of a mirror and act out the animation yourself if at all possible (you can also ask someone else to act it out for you and even recorded it for later reference). To ease the explanation we will go through a ridiculously easy animation. Begin by selecting the neck bone with the [RMB] and press [I], then select the Insert Key >> LocRot option. You will now see a key inserted under the neck bone in the Action Editor. Now advance the frame count to 10 by pressing the [RAB] (right arrow button).

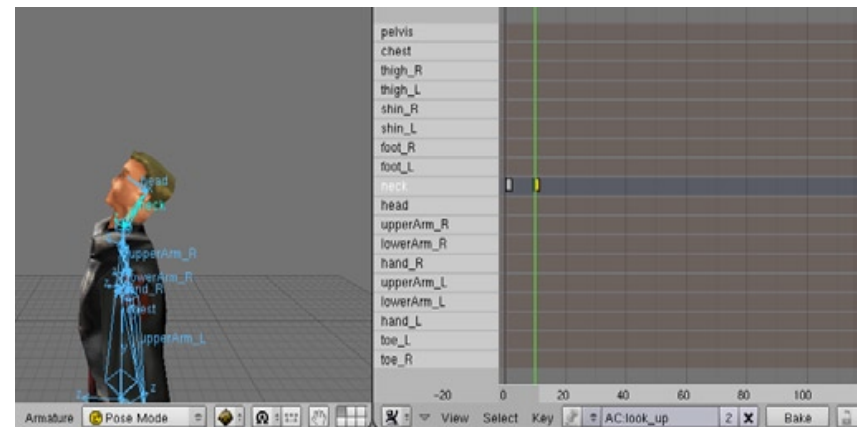
Rotate the neck bone about 30 or so degrees (making him look up) and insert another key like before. Bring the frame count back to 1 by pressing the [LAB] (left arrow button) and press [Alt-A] in the 3D View. If done correctly, you should see the mesh look up towards the sky. See image on the right.

Now move over to the game

engine buttons or Logic [F4] buttons. Create exactly what you see here (notice this action or AC is called “look_up”). See below. With that setup and your



mouse over the 3D View, press [P] and then [ENTER] ignoring the no (correct) camera error message. If done correctly, you should now be able to press and hold [L] making the character look up. For further practise create a hand waving animation, an idle animation, and a walking animation.



Final thoughts.

For video games, most animations will be simple and in place or in one location, for instance, the walk animation will be the character walking in place. It is not until you bring the animation into the game engine will you advance the mesh position while playing the animation to create the illusion of walking.

For in game movies or FMV, more complicated animations will be needed obviously. Remember that the more frames you have between KeyFrames will make the animation slower or faster if there are less KeyFrames. If, while animating, you find the bones to be too twisted or hard to work with than it may be easier to clear the rotations for that frame and position from there.

Every effort has been made to state correct information, however, if you find anything in error please contact me. Note, that I do not claim to be an expert so if you feel something should be done differently than go ahead and do what you feel comfortable with, for I'm only providing this information to help rather than dictate ■

LINKS

You can find the accompanying .blend file at <http://www.cgartwork.com/dcl/characteranimation.blend> and an AVI movie of a more complicated animation than the one we did here at <http://www.cgartwork.com/dcl/characteranimation.avi>.



David Lettier is a “hobbyist” game developer practising and honing his development skills since 2004. Being a “lone wolf” developer, David Lettier is a “home grown” programmer, animator, and sound designer. If you would like to contact David to drop a comment/question/suggestion and/or to join his cause you can reach him at dlettier@gmail.com

Rigging a Spider in Blender

-by Gabriel "gabio" Beloin



Some basics thoughts before we start

We are going to rig a real-looking spider. The best way to start is to get some real footage of our goal first.

- * To look like a usual spider having 8 legs.
- * The body is made of 2 rounds parts
- * The head has 2 mandibles
- * The whole thing is symmetrical

We will build a rig for this little thing. During this process it's important to keep in mind that our goal is to build something easy to use and that can give good looking animations without too much work. It's important to think about that, because you may well be the one who will use this rig. It must be simple to use yet give a lot of possibilities.

Building A Rig Just For Him

Now have a look at the blend file (included w/ this issue of Blenderart). There is a spider I modelled for the purpose of this tutorial. We will use it as an example.

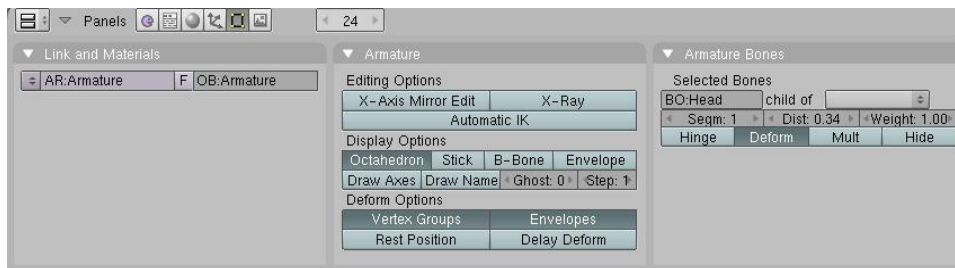
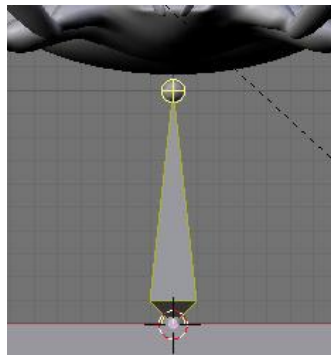


Start By adding an armature to the scene.

ADD >> Armature.

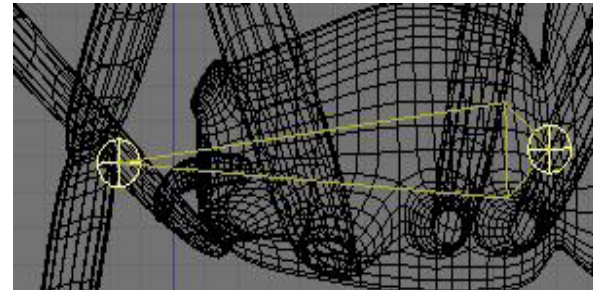
Now you are in Edit-Mode, editing the new armature you just added. There is a bone where the cursor is. Select all of the bone by right-clicking on the middle of it (the body). This will select both the root and the tip of the bone. You can then move it around.

The bone is made of



the root which is the lower ball, the body is the octahedron, the tip is the upper ball.

move and rotate this bone around so it's in the head of the Spider, the centre of the character:



This bone will have a name, we will call it "Head". To name a bone, go in EditButton [F9] There you'll see some panels useful to us.

What we are looking for is in the "Armature Bones" panel. If the Head bone is selected, you'll see a bunch of buttons under "selected Bones". There the "BO:" field will let you change the name of this bone from "Bone" to "Head".

To see the result in the 3DView, toggle the "Draw name" button on. It's in the Armature panel.

We will continue to add bones then.

In side-view[3NumPad] select the root of the “Head” and extrude a new bone with [E]. Now you have two bones and each of the roots are at the same spot. Name it “Abdomen”. Make it a child of the Head by going in the “Armature Bones” panel with the “Abdomen” bone selected, clicking on the drop down on the right of “child of” and selecting “Head”.

It’s already time to start to think about the hierarchy of your rig (example on right side).

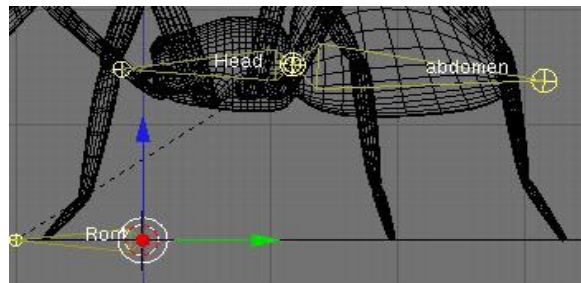
You can parent a bone to another one. This means the child bone will do what ever the parent does. This gives you the ability to build solid rigs. If for example you move the body, all the legs will follow. It’s important to know what will depend on what. A general way of building a rig is to start with a master bone. This bone will be the parent of all the others. This is important in case you want to move everything, you will only need to



move this bone. This master bone is generally named “Root”. Add a new bone while in editmode (Add >> Bone), name it “Root” and put it somewhere away from the character (at the bottom is a good place).

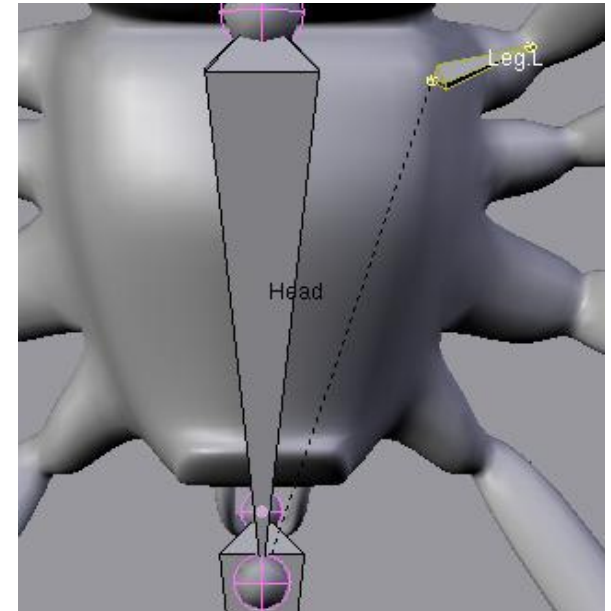
You can parent the “Head” to the “Root”, but there is a faster way. Select the “Head” bone, then the “Root” bone so “Root” is active, then do [CTRL-P] and select “Keep Offset”. Connected is another type of relationship. It’s basically a parent-child relation, but the root of the child will follow the tip of the Parent. This will let you create chains of bones like the bones in your arm. If you raise your arm, the rest also follows, doesn’t it?

So we now have something looking like this:

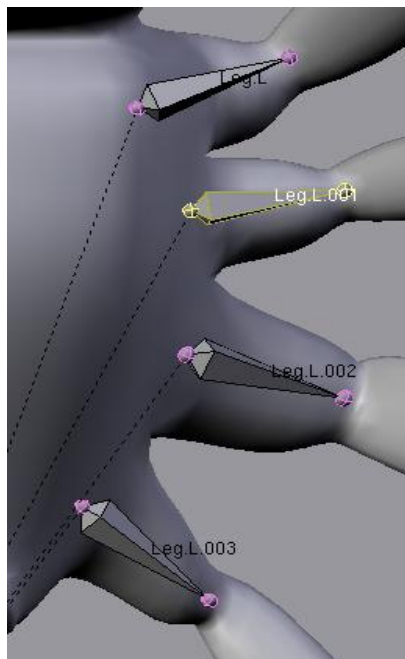


We will now rig the 8 legs. Don’t worry, there is a way to do just half the job. Since the Spider is symmetrical, it’s possible to tell Blender to rig the other half while we do our part. Go into the Editbutton panel and toggle “X-Axis Mirror Edit”. Go into side view [3NUMPAD] Select “Abdomen” and Duplicate it using

[Shift-D]. This will give you an other bone with the same parent as “Abdomen”, “Head”. Move it to one of the left legs so it’s placed where a leg is plugged to the body and the tip is in the middle of the first joint. Rename it to Leg.L:



Make sure the bone is in the middle of the leg in side view too. Now Using [Shift-D], duplicate this bone for the other 3 legs. Do not touch the right side.

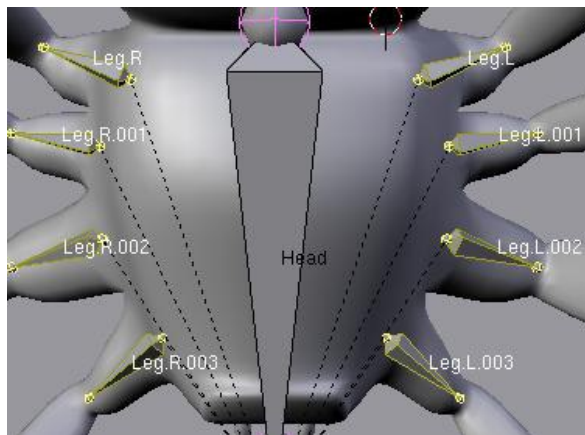


Now, for Blender to mirror our moves on the X-Axis, there must be 2 bones following the name convention. The name convention will let Blender know what is the left part and what is the right part of a rig. The names must end with .L .l .left .Left or .R .r .Right. right. If there is

number after (like “name.L.001”), it’s not a problem. So if you have a bone on the Left named leg.L and a bone on the right named leg.R, Blender will mirror our move for the matching bones.

For this to work we will copy the 4 bones we did and give them proper names. Select the 4 Legs’ bones and duplicate them. Move the duplication away, it doesn’t matter where. Press [W] and select “Flip Left-Right

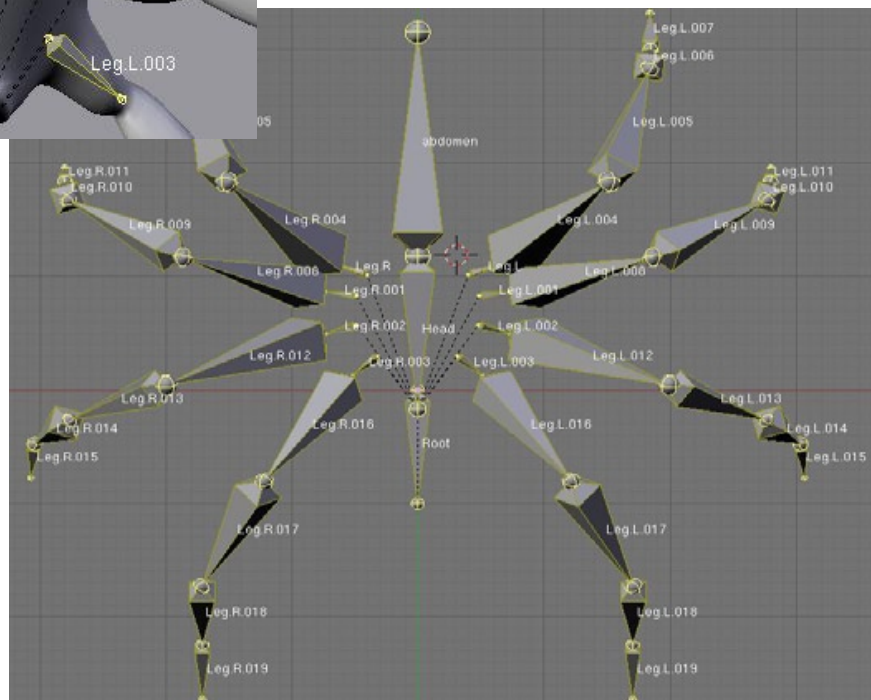
Names”. This will change the names from Leg.L to Leg.R. Re-select the 4 original bones (.L) and move them a little. The 4 new bones will get the mirrored position. Nice isn’t it? (Image below).



Now you can start extruding all the bones on the left to fill the legs. There are 4 more bones needed by each leg, make sure the bones are in the middle of the geometry by looking in side-view and top view. Each joint should be where the geometry is more narrow. The names are automatically generated and are OK. As you go, you’ll notice Blender will copy your moves and build the other side of the

armature with you.

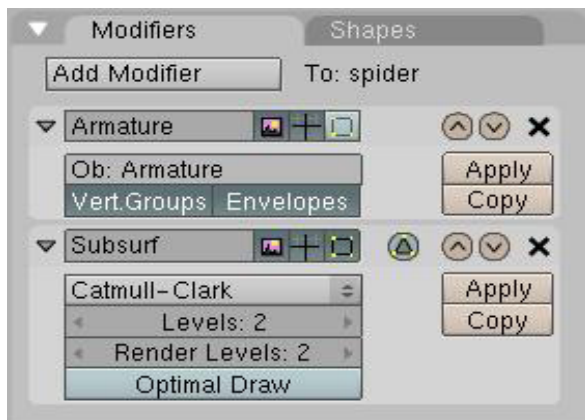
When done, recalc the rotation of all bones to be sure they are facing up. To do so, select all the bones and do [CTRL-N]. The result should start to look quite complex, see the image below.



The Armature Modifier

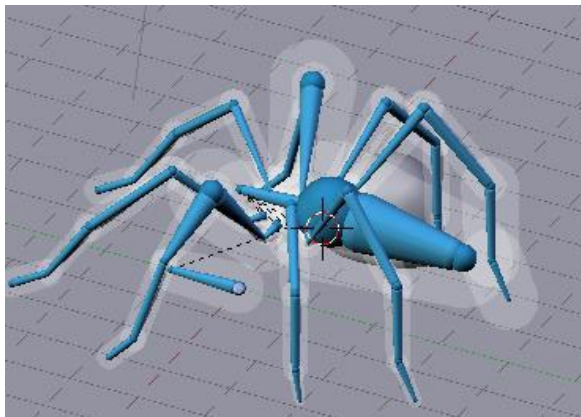
When you are done with that, we can start looking to deform the mesh using the Armature. To do so, exit EditMode with [Tab] and select the Spider. Go into the EditButton [F9] and look for the Modifier panel.

In the Blend file there is already a modifier, the SubSurf. Press the Add Modifier button and select Armature in the list. Look for the “OB:” field and type in “Armature”, which is the name of the armature object.



Now when you select the rig and go in pose-mode [CTRL-Tab], moving a bone should move something of the Spider. But in general it's going to be far from perfect. This is where we start tweaking envelopes. Go back into the EditButton, in the Armature panel, there will be a “Envelope” toggle. Turning it on will let you see and tweak those envelopes. Basically they define an area for each bone where the geometry will

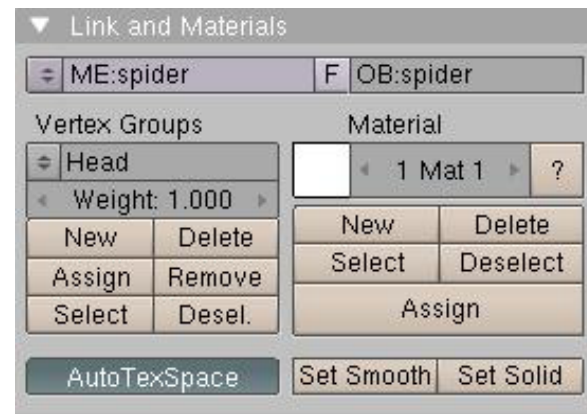
follow the bone. Changing the area to better match what we want is done by selecting a bone and scaling the zone with [ALT-S]. It's also possible to tweak envelopes in EditMode of the armature (by pressing [Tab]) and there is generally more options, like scaling the tip or root ball. At first the envelopes look like this:



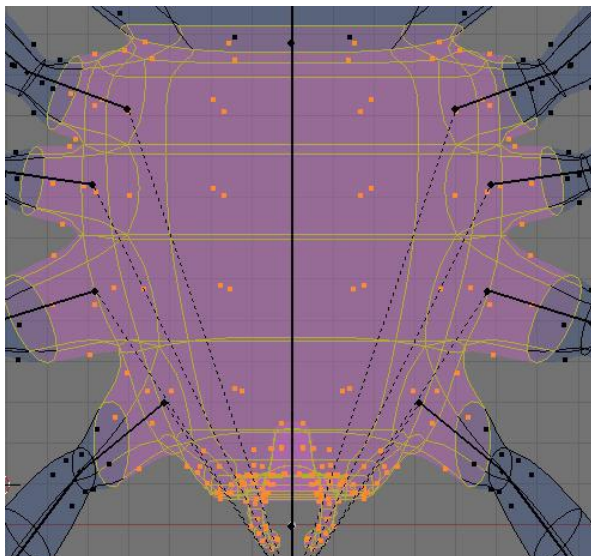
Tweaking the zone to touch all the geometry yet blending each zone into each other nicely is hard. It's possible to visualise the result of scaling the zone in pose-mode: Move a bone, if there is some vertices not following, scale the zone a bit until you see the vertices move where they should be with the rest of the gang.

Some bones like the “Head” and “Abdomen” will not be feasible using the envelope option, so we will need to do it by hand. Just scale it to 0, so we don't use the envelope for this bone. There are 2 ways for moving vertices: Envelope and vertex groups.

When you are done with the legs. We will make sure the “Head” and “Abdomen” work. For this select the Spider and go into EditMode. Go into the EditButton and look for the “Link and Materials” panel.

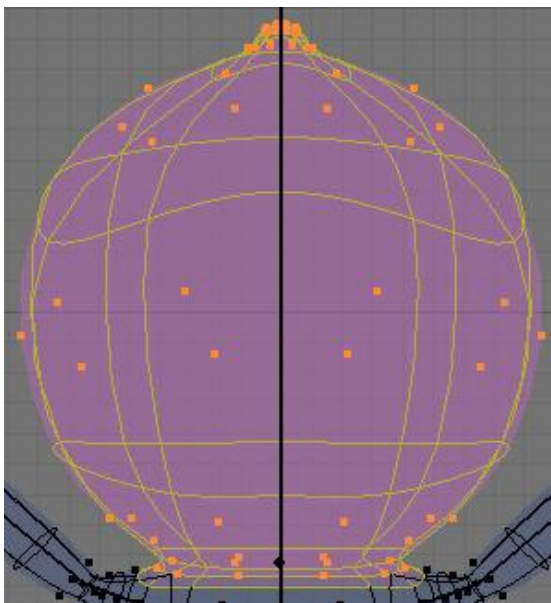


There you can add a group by pressing the New button. Name it “Head”, the same name as the bone. Blender will naturally match those two and will use this vertex group when you move the “Head” bone around. Select all the geometry in the head like this: Press the ‘Assign’ button. That’s it. The Weight



number is the weight you’ll give to this group when assigning. For now a constant weight of 1 everywhere is enough. Do the same for the abdomen:

Test it by moving the armature around, all parts of the spider should follow. In case it doesn’t, assign the missing vertices to the proper group.



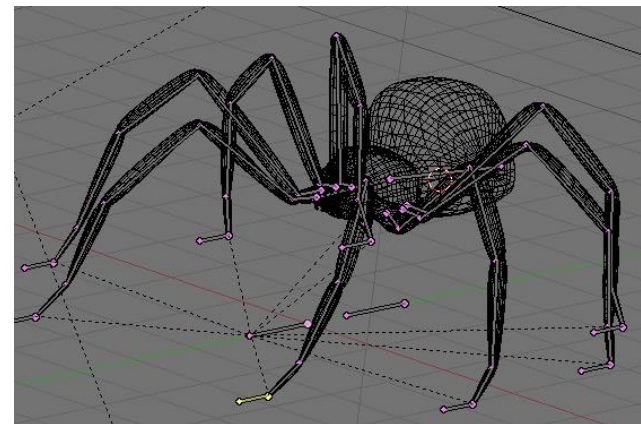
Automate Some Tedious Tasks: Constraints

Now it would be possible to keep it this way but it would be rather long to animate each bone of each leg. Let add some magic to the pieces.

We will add a IK solver to all of the legs. This will let us animate a complete leg by moving just one bone. Select the Armature and go in EditMode. Select the tip of all the last bones of each leg, go in top view and extrude a new bone from them. With all of the new bones selected, press [ALT-P] >> Clear Parent. Then re-parent all bones by selecting one of the new bones, then select the “Root” bone and pressing [CTRL-P]. You will have to re-parent each bone one by

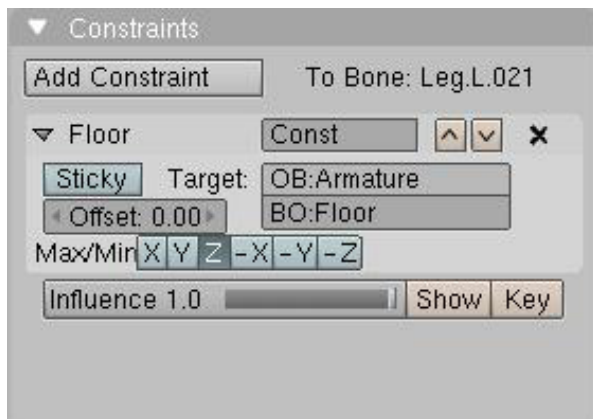
one, sadly it doesn’t work when there is more than two bones selected.

Just before adding IK solvers all over the place, it’s wise to put all other identical constraints and copy them all around. It’s a time saver. We will add a floor



constraint to one of the newly extruded bones, then copy this constraint all over the place because it’s identical (all constraints target the same bone [the floor bone]). Then we will be able to add each IK solver one by one, as they each have a unique target. Doing it in the wrong order would screw it all up, since copying constraints erase all the constraints that were already there. So always create and copy simple constraints (such as the floor constraint), then add special constraint (such as IK solvers) one by one after ward.

To do so. Add a new bone by going in side view and duplicating the “Root” bone, place it flat on the ground as if it was the ground. This new bone will not be parented to anything, and it’s better like that. Name it Floor and go in pose-mode. Go into the Edit-buttons and press the “Add Constraint” button --> Floor. It should look like this:



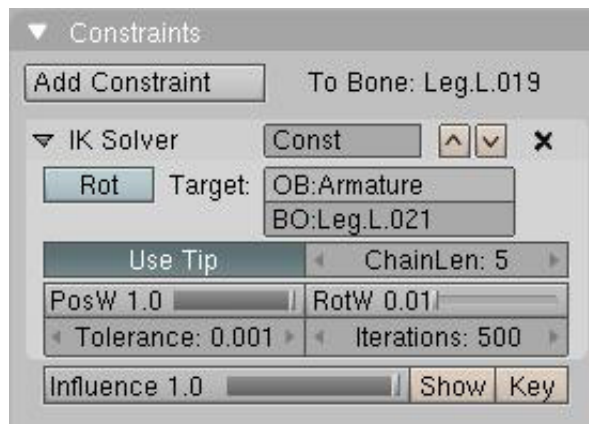
Tell Blender that the OB: to use as target is “Armature” a new field will appear, BO: enter the name of the bone you want to use as floor, here we made the bone “Floor”.

Time to copy this to all the newly extruded bones. Select them all and selecting the one you added the Floor constraint to last. Do [CTRL-C] in 3DView --> Constraint. Tada!

One last thing, with all the newly extruded bones selected, go into the EditButton --> “Armature Bones”

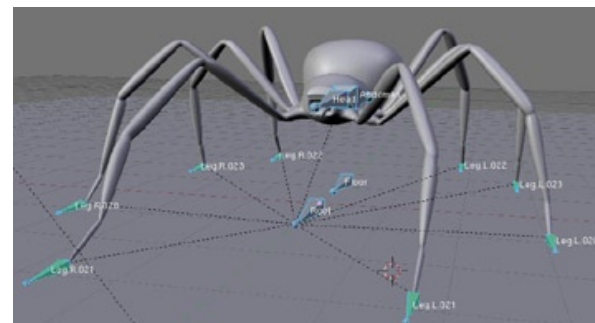
panel and toggle all “Deform” buttons off (one for each bone). Do the same for the “Root” bone and the “Abdomen” bone, since they will not move any vertices at all.

Now the IK solver. Go into pose-mode [CTRL-Tab]. Select one of the new bones you just extruded then the last bone from the leg you extruded the new bone from and press [CTRL-I] --> to selected bone. This will add an IK solver constraint to the leg’s last bone targeting the new bone you just extruded. If you try to move the target now, you will notice the entire body is moving and the orange dotted line go to the “Root” bone. This is because the IK Solver goes up in the hierarchy until it finds the top. To limit this, you can use the “chainLen:” number button. It’s located in EditButton --> Constraints panel in the IK Solver Block.



Setting it to 5 should point to the root of the first bone of the leg. Repeat for each leg.

The result should let you pose this little Spider in an easier way. You only need to move one of the IK targets to move a leg. You can also move the head and all the legs will move accordingly. It’s possible to raise the “Floor” or move the “Root”. According to this, we can hide all the useless bones. Select all bones you don’t need when animating and press [H] The result should be like this: ■





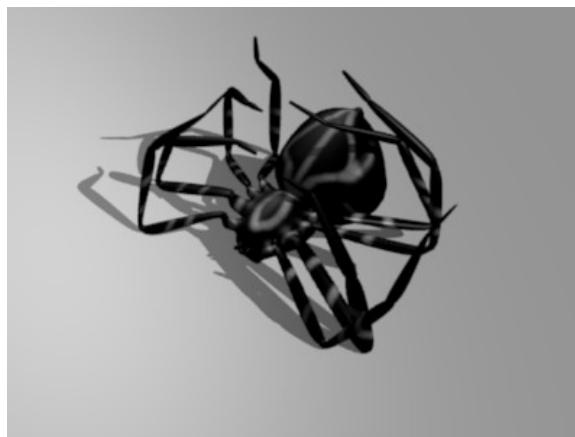
Climbing a wall...



Annoyed by some thing...



Attacking...



Ho he is dead!

NOTE:

Check out the blend files at blenderart website for this tutorial. One of them contains pre-made poses like the ones at left.



Gabriel Beloin (gabio) is from Sherbrooke, Canada. He is now working as a network administrator. Gabriel used Blender for 3 years, since version 2.26.

Active in many communities such as Elysiun, Zoologique and CGtalk, he also take part of the Blender development process as python coder and beta tester.

Making of: The Age of Steam

- by Zsolt Stefan

Concept

All good pictures have some story behind them. I got the idea for this machine from a real one that I saw in a local 'historic park', essentially an huge open air museum. There it was, a huge beast of iron, all rusted and old. But you could tell that when it was made, it was worth a lot more. It is a steam mobile, basically a steam engine, placed on wheels, so you could transport it wherever it was needed, and then run other machines off it. Already when I saw the thing, I knew that I wanted to make a 3D render out of it, so I made the best of the situation, and made two photos, one of them exactly from the side. Plus two more close-ups of the metal covering. These later came in handy as references, and textures.

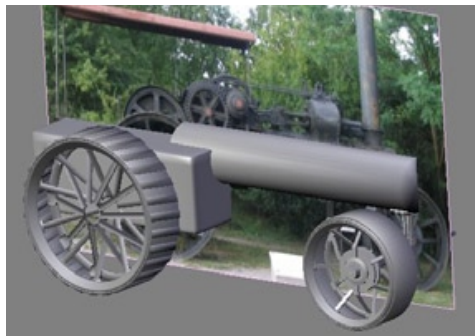
Modelling

It was fortunate, that I took a side photograph of the engine, making the modelling easier. Here is an earlier version of the blend file, during the initial modelling phase: See the pic on right.

There is a plane in the middle, with the picture I took placed on it with UV mapping, so that it shows up in the 3D viewport. This plane is scaled to the correct height and width for the engine. A very important step before starting modelling is determining the scale. Blender uses its internal units, not any real

world units, so to make a realistic scene, you must decide what 1 Blender Unit (BU) means. In this case I went with 1BU=10cm, the whole engine is 4 meters long and 2.4 high.

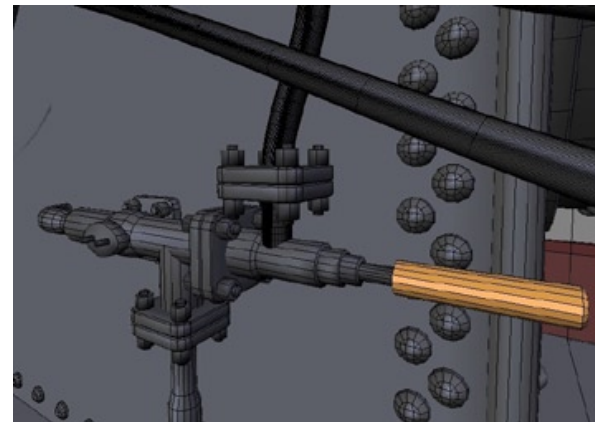
Modelling is mostly starting with a plane or cube, and then going from there, with extrude, subdivide, and the loop modelling



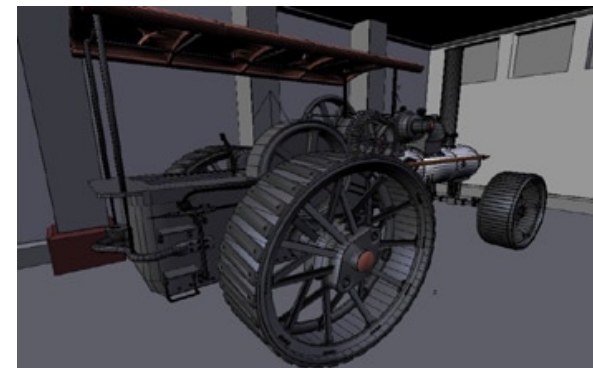
tools. All of this with the help of the UV mapped plane in the middle. As most of the engine is mechanical models, I barely used any SubSurf, so all of the detail is modelled by hand. A useful little script that I often used is the Bevel Centre. This allows me to add little bevels to all the objects, as there is no such thing as a totally sharp edge in real life. Bevels catch the light, and give it more realism, and also add to the wear and tear. But be careful! Only add bevels when the model is finished. There is no way to undo them, and you can't really change or add to the model once the edges have been bevelled.

Details... Very important to add that 'final touch' of realism to any scene: adding the small details, these are what make it real. Here is a close-up of one part

of the model, with the wires visible. Note the nuts and bolts, the connections are all modelled. Some modelling, like the parts at the back of the engine can't be seen in the final render any ways...

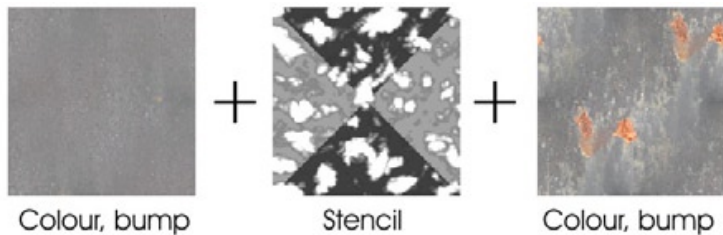


Another view of the engine, from the back:



Materials, textures

During modelling, I already start adding materials and texturing. Most of the metal parts use one material, a rusty metal. This was made with a little technique that I use to hide repeating textures. The way it works is: there are two metal textures, both made seamless, one is a simple metal, the other one is a rusty metal, similar in colour. These two textures came from the close-up photos I took. They are separated by another texture channel, a clouds texture, with colour-band turned on. The colour-band is half white and opaque, the other half is white and transparent. This is then put between the other two textures, and the 'Stencil' button is turned on in the Material settings, Map To panel. This makes some parts of the rusty texture to show through, and some not,



randomly. The resulting material. (Image left)

Then the Map Input settings of the three textures is adjusted

until the place where the rust shows up on the model is satisfactory. A similar technique was used for the walls in the background, as well as the floor.



The scene

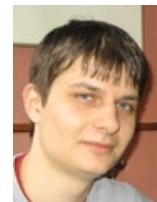
From the beginning, I envisioned the large machine left to itself in a warehouse of some sort, with just a little light shining in. The following is a preliminary render of the room. I added just a few objects to the scene, to give a sense of scale and realism, but not too much to draw the attention away from the large steam mobile in the middle. The room itself is simple, with a few columns at the back wall, and some windows: on the left wall, the ceiling, and the wall behind the camera. These are the ones casting the shadows. As a small detail, the windows are broken

The lights: there are three lights on the outside of the room, behind the camera, shining through windows, casting shadow mapped shadows. Plus Yafray's global illumination was used, this added the realistic spills of light, for example around the windows at the top left,

though increasing the render time several times. The camera was placed low, pointing upwards a bit to convey a sense of a large machine. The strong contrast between light and dark also added a definitive mood to the scene.

Final render, post-pro

I did the final render as an A4 @ 300 DPI, 3636*2657 pixels, this took about 15 hours. As a finishing touch, I loaded it back into Blender, and used the glow filter in the sequence editor to add a bit of glow to the lightest parts of the image, eg. around the windows. Then some post-pro colour touch-ups and sharpening resulted in this final image ■



I live in Hungary, and am currently studying industrial design engineering at the Budapest University of Technology and Economics.

Some of my hobbies are: 3D graphics of course :) , concept design/product design, plus watching good movies, going out with friends, hanging out on elysium and other forums, drawing."

The Plumíferos Project

-- Claudio "malefico" Andaur

Introduction

In this article I would like to briefly summarise the realisation of the trailer/teaser of the "Plumíferos" Project, a commercial full length 3D animation film project we are making in Argentina using Blender. This trailer was first screened during the Blender lectures at Siggraph 2005 and later on was nominated for the 2005's edition of Suzanne Award at the Blender Art Festival. If you couldn't watch it, pay a visit to <http://www.plumiferos.com>.

The film's storyline is about a little male sparrow named "Juan" and a female canary named "Feifi", and their journey to find themselves. It is aimed mainly for a children audience with lots of hilarious gags and psychological background for the grown-ups.

This project started around a year ago, when director and original script author Daniel DeFelippo came to Gustavo Giannini from Manos Digit ales and me with the idea of making a 3D animation film. Gustavo and I were already using Blender and other Open Source tools to create CG animation and VFX for prime time television shows, TV commercials and also for teaching 3D character animation in a local digital art school.

Figure: Visual FX made by Manos Digitale Animation Studio for Pol-Ka Producciones (all Blender 2.36!)

Daniel's 3D experience was mainly based on 3D Studio MAX and he was quickly convinced by us (bwahahaha) to switch to Blender as a main animation tool. We

and weekends. Organisation was one of the first and hardest problems to solve.

Daniel's original characters and designs were modified, modelled and textured at the same time that sets were created to resemble certain parts of Buenos



soon gathered a team of local Blenderheads (most of them former students of ours) and started making a trailer with the characters created by Daniel.

Even when we weren't a big team, it soon became apparent how hard it would be to coordinate every body's work and assemble all parts of a every single shot. Sometimes a texture was missing, on others there were two versions of the same model or different lighting set-ups for sequential shots. Besides, all of us had (and still have) full time non-CG related jobs to attend and that forced us to work in our spare time

Aires city where action occurs. There were many issues to address, we had characters with feathers on their wings, lots of trees, many cars around, and motion blurred wings, plus Blender's own limitations and caveats. Keep in mind that we made it all using Blender 2.36, so we had no fancy character animation tools available at that time.

Some technical stuff

Since we didn't have many fast machines available for rendering, all known tricks were used to speed up render times. For instance, arrays of negative lights were used to simulate diffuse shadows or ambient occlusion, some objects were rendered separately in layers, others were replaced by textures, etc.

For trees we used both L-System script available at Blender.org, and Arbaro (<http://arbaro.sourceforge.net>) the excellent Java app. We made several renders of a tree with different light orientation and then used these renders as textures for planes in order to keep poly count low. Other common effects like depth of field and atmospherically haze were cheated by using B/W masks of objects and composing in the post production stage, a trick well known by average Blenderheads. About this, we used the Gradient Method (you can read about this technique at: <http://www.elysiun.com/forum/viewtopic.php?t=30959>) for creating the B/W masks instead of typical Z-Buffer plug-ins, this was done this way for several reasons, we had several objects with alpha textures so if we'd have used any Z-Buffer plug-in, those objects would have been rendered as invisible objects.

Motion blur of wings was trickier to do. Blender's motion blur was too slow to be practical in many situations, plus we wanted to blur only on Juan's wings, not on Juan itself, so we had to make a special version of our little bird with only its wings as renderable parts. Then several layers were rendered by manipulating the NLA strips of the wing beating to get a more natural look of the blur. Finally, all wing layers were blurred and composed together with the layer of the bird without wings. Additional motion and zoom blur were added over the final frames in composition software. Blur of cars wheels were made in a similar fashion. Wheels spin was done using a frame driven python script.

Feathers were generated by a python tool written by us which let us vary the size of the feathers following a designed contour. Every feather had a two-bone armature which let us make feathers bend if needed. Each feather had several copy rotation constraints which were controlled by key bones in main character's armature, so we didn't have to care about every feather at animation time; we only animate the key feathers of every wing. As a final step, feather's armatures were "Vertex Parented" to specific faces of wing's mesh, in order for feathers to follow character deformation correctly.

Juan's acting in "audition" scene was done using Blender's NLA features and RVK. All open/shut movements of wings were animated previously and combined with remaining corporal animation using NLA. Our feathered wing set-up worked perfectly with this scheme. The usual work flow for animating this scene was first making the corporal animation, then the facial animation and as last step the lip-syncing.

Argentine professional actors Guido Kaczka and Manuel Vicente gave their voices for both bird character (Juan) and "off" voice of director in audition clip. Guido's particular performance was key to inspiring the facial and corporal acting of Juan. In our pipeline, we found it extremely useful to film actors during their performance to imitate some of their involuntary gestures.

Blender's sound scrubbing feature was extensively used to get the proper lip-syncing animation of Juan. Sound editing and music were done by the Sonomondo sound studio once all sequences were properly edited.

A few notes on work flow

For the "highway" scene, we worked first on a "proxy" version, using very simple objects representing the differ-

ent characters and cars. We focused our work at this stage in animating objects and cameras through paths. The complete sequence was split into several shots from storyboard, and using these proxy scenes, we made a 3D animatic, which we rendered in OpenGL.

Once object and camera animation were done and corrected, proxy puppets were replaced by final characters. This process made dealing with large scenes a lot less painful. Actually, main characters were added on a separate scene while all heavy-loaded backgrounds remained as a separate set. By doing it like this we could focus on flight choreography and gesture without having to care about real-time transformation of heavy background geometry. Besides, at the same time we were animating, the characters were being modelled and textured separately.

Render strategies needed to be carefully designed from day one. It is very important to have a clear picture of what can or cannot be rendered separately and what can be done in post-processing instead of trying to render everything from the start.

Future work and plans

The making of these videos was not only a fun and exciting experience, it was important to realise Blender's own capabilities and limitations. Most of the problems we found during this production were reported to the Blender coders team and were being worked on together with Orange Project's own findings.

We strongly believe in Blender's potential and power, and the recently added animation features only demonstrates that we weren't wrong when we chose Blender as our main animation platform. Moreover, during the making of these trailers, we also used Blender for making of visual FX in local TV shows in prime time, TV commercials and other professional applications with great success.

There are still many issues to overcome to reach our goal of making a full feature 3d animation film, for instance we are in the process of setting up a real Studio to gather all artists so we can work together in a creative environment.

Acknowledgements and thanks

Of course none of our dreams could become true if it weren't for the continuous and excellent work of the Blender's coders team hunting bugs and developing new features. We can't thank every one of them enough for making such a pro tool as Blender.

I would like to thank Chris Want, Roland Hess, and

Martin Poirier for their help and advice, and of course Ton Roosendaal for just being Ton !

A special acknowledgement should be made from my part to all the Plumíferos Team and all the people who gave their time, effort and even the lending of their machines to render expecting nothing but the Project's success, in no particular order:

Manuel Pérez, Diego De Gennaro, Iván Hoffmann, Raúl Medina, Alejandro Am, David Teres, Alvaro Vidal, Claudio Dobniewski, Felipe Sánchez, Lorena Maza, Pedro Knigge and Ale Barbesi ■

The Plumíferos Team

Director: Daniel DeFelippo

Executive Producer: Gustavo Giannini

3D Supervisor: Claudio "malefico" Andaur

Animators:

Manuel Pérez

Daniel DeFelippo

Iván Hoffmann

Claudio Andaur

Models, textures, lighting:

Iván Hoffmann

Diego De Gennaro

Raúl Medina

Claudio Dobniewski

Alvaro Vidal

Alejandro Barbesi

Claudio Andaur

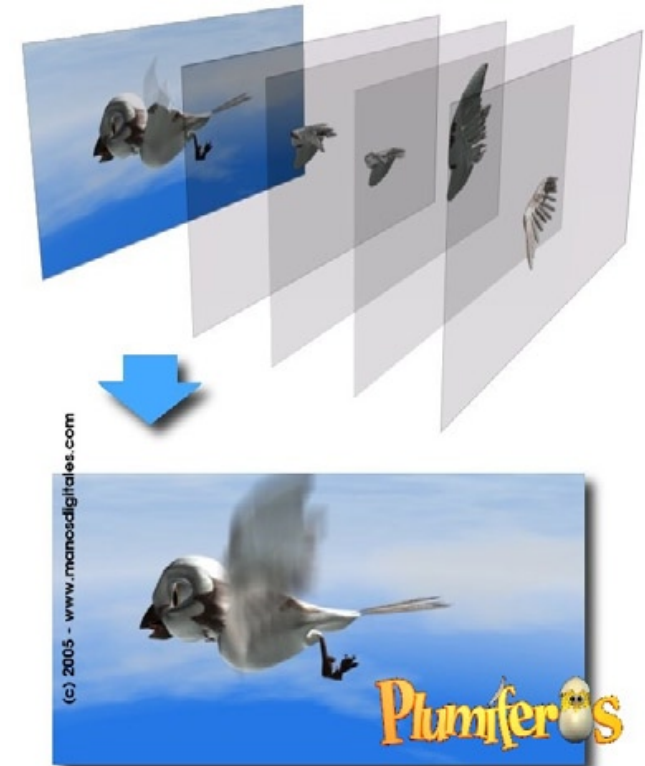
Python scripting:

Claudio Dobniewski

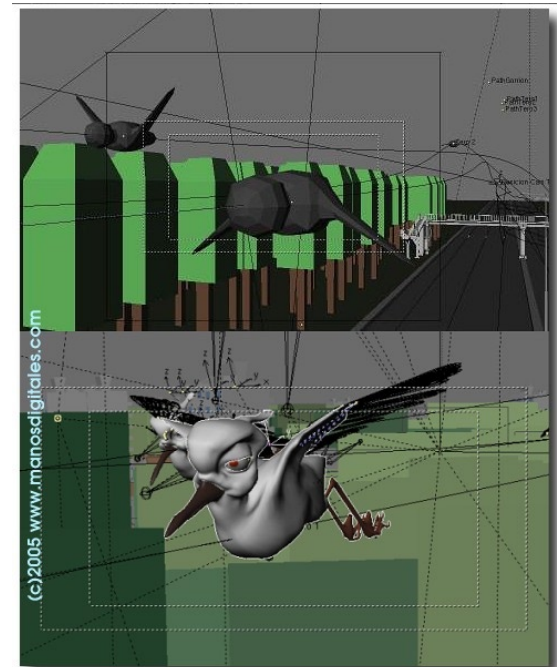
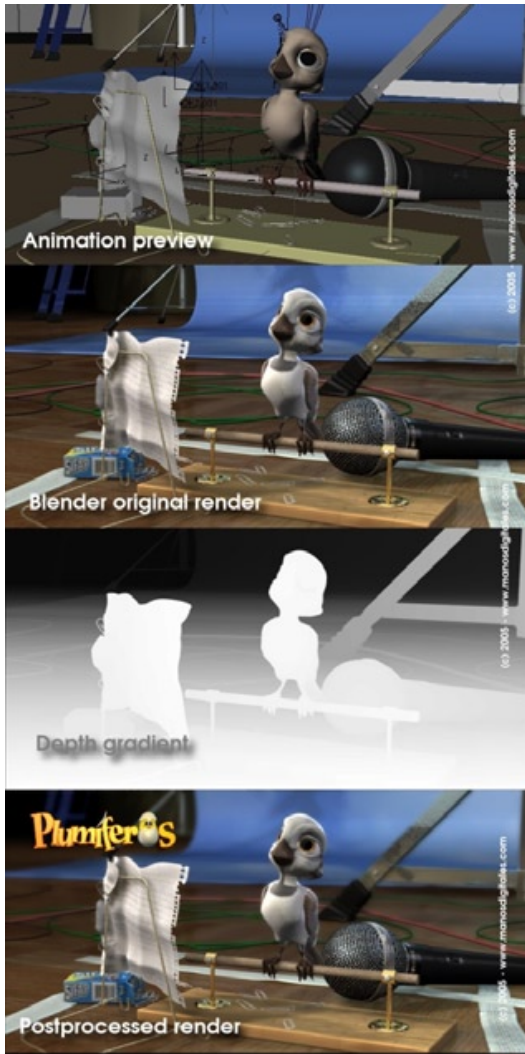
Claudio Andaur

C/C++ programming:

Felipe Sánchez



More pictures on next page...



Claudio "malefico" Andaur

I was born in Buenos Aires, Argentina in 1970 (man, I feel old!) I'm a chemical engineer and working with Blender since 1.80. I teach Blender in a Digital Art School in Buenos Aires, and have written Blender documentation both in spanish and english (been coordinating the Documentation spanish translation effort and traslated several docs available at mediawiki. blender.org). Made several Blender presentations in linux communities and universities from Argentina and other american countries. Currently I lead the 3D and character animation of Plumiferos Project. I have a lovely little baby girl, and a marvelous wife.



Making of : New Penguoen 2.38

--by Enrico Valenza

Introduction

The goal of “New Penguoen 2.38” was to try to reconstruct with Blender the wonderful 2005 Citroen C4 spot (the dancing transformer’s one, made by Embassy VFX).

I loved that spot as I saw it, so I had the idea: wouldn’t have been funny to replace the giant robot-car with a giant silly penguin? In the meantime, the Suzanne Awards deadline came out, so I thought to submit it, if only it could be finished in time.

The gag of the roasted penguin at the end of the animation came out in progress; so, as the headline of the original spot was “New Citroen C4 - Alive with Technology”, the headline for my parody became “New Penguoen 2.38 - Cooked with Blender 3d”. 2.38 was the version number of Blender I used, with all those new marvellous animation features!

From a few researches I did, it came out that the C4 spot was located in Vancouver, Canada, on the roof of a post office. Actually, Embassy crew shot high resolution photos of the environment and then re-mapped the images on 3d geometry. That’s exactly what I did inside Blender: the only difference was I hadn’t the hires photos of Vancouver, so I had to paint in The Gimp detailed textures of the buildings and of the far

landscape. But first, I had to set the environment.

The environment

There are a few python scripts that can be used to have camera-tracking in Blender; I preferred to do all the work by hand. I didn’t want a too perfect matching. I loaded a low resolution AVI of the original spot (found on the web) as background in a camera window, to use it as reference (a template, if you prefer).

OK: first, add a plane to your scene (*Add>>Mesh>>Plane*), then assign a material (F5); go in the texture window (F6) and load the animation as Image, selecting the Movie option. Then put the mouse cursor on a 3d window and press NumPad0 to have the camera window; in the window header go to *View>>Background Image>>Use Background Image>>Load texture* and load the texture of the material.

At this point you can delete the plane, its only purpose was just to have a texture to be selected.

In this case, the background AVI worked as a storyboard too; I had all the shots and discovered that the total length of the spot was 751 frames.

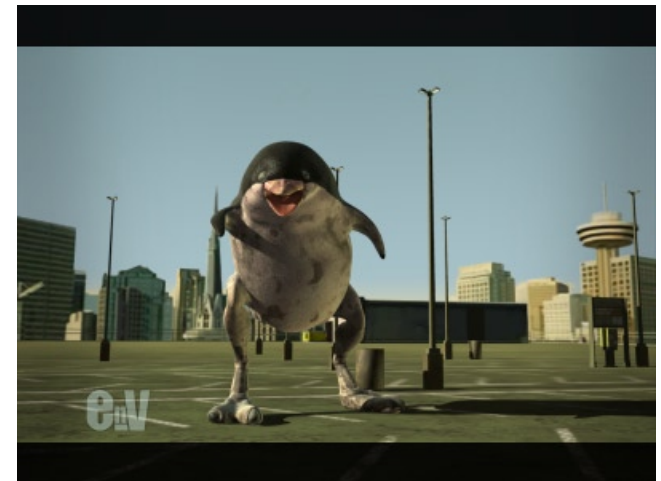
I started to add a simple plane for the floor (actually the roof of the post office) and some cube primitives for the main elements.

Looking in the camera window, I moved the

camera (select the camera, G) to match the perspective. With a few tests I found that the focal length was near to 31.0 (Lens value, F9; *I>>Insert Key>>Lens* with mouse on the camera panel to animate).

This way, by adding IPO’s keys to the camera position and rotation (*I>>Insert Key>>LocRot*) and always looking for the best correspondence between the original spot elements and my cg primitives, I made all the shots and at the same time I restored an almost similar environment.

By putting a sphere in the centre of the scene and making little renders, I matched also the direction and the intensity of the sunlight. A lamp under the floor provided a fake radiosity effect.

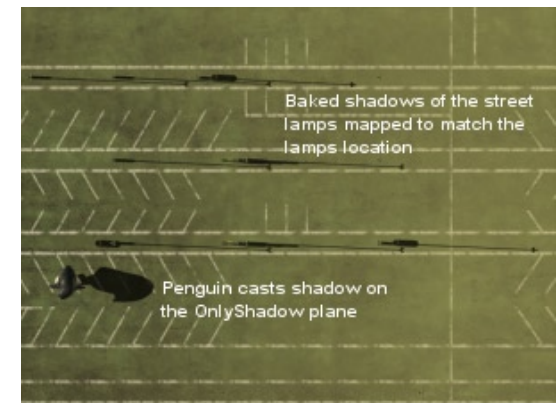
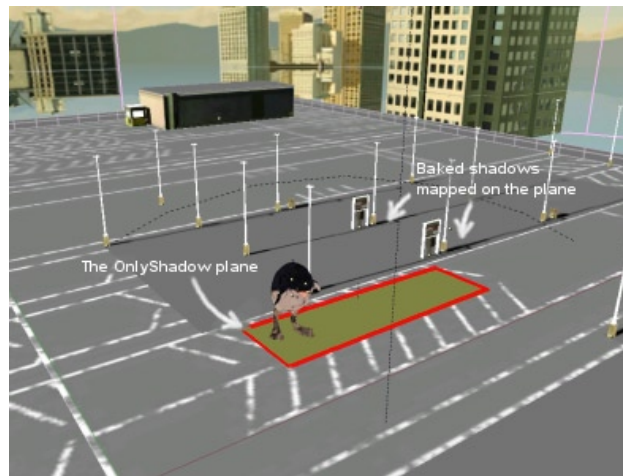
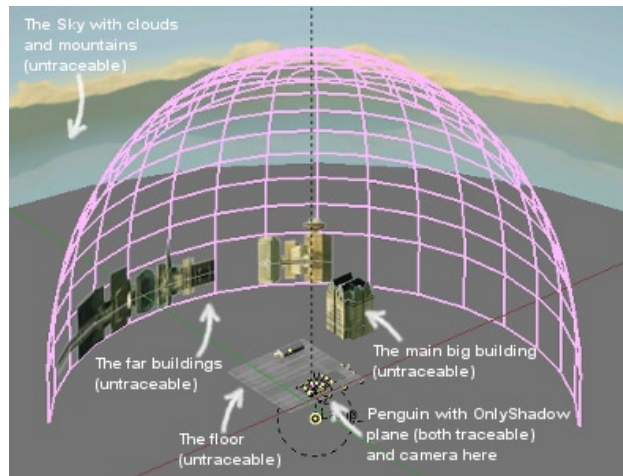


751 frames may take a very long time to render; ray tracing lamps casting shadows add a lot of time too and if you have a very big setting, the octree (bottom of render window, F10) must calculate very far objects. In the end the rendering times become huge, not the best for animations. So my priority was to lower rendering times in every possible way.

A way to do this is to let the octree calculate only the main or the moving elements in the shot, by setting the materials for the motionless and far elements to shadeless and de-selecting the traceable option in the material window. The octree won't calculate the untraceable materials and, as they are shadeless, there are no shadows to be calculated.

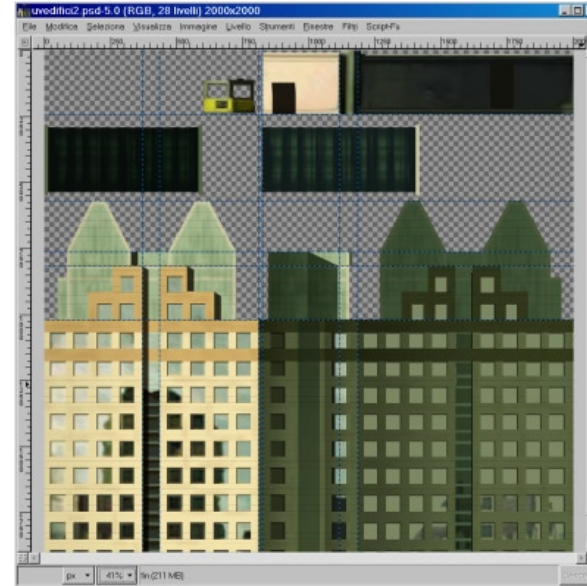
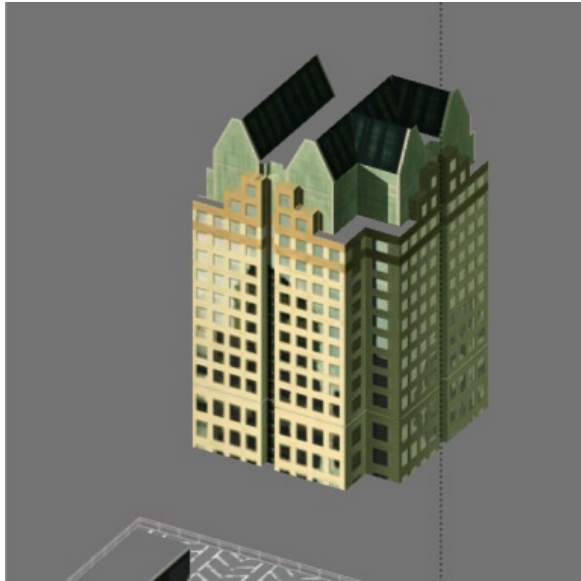
But what about the penguin casting a shadow on the shadeless floor? A small OnlyShadow plane has been constrained (CopyLocation constrain, Constraints panel F7) to an Empty parented (Ctrl+P) to the armature, so to follow the penguin movements. I deselected the Z axis of the constrain, so the plane didn't rise as the penguin jumped.

Of course, every shadow and light effect for the shadeless objects had to be painted in the textures, or baked in some way.



The casted shadows of street-lamps and baskets have been baked by a rendering from above, and the alpha channelled image has been mapped on an untraceable plane to match their location.

The main building on the left mid ground has shade-less materials as well, but has been modelled (only for the visible sides) just to increase the perspective effect while the camera is moving. Then, it has been unwrapped with the ArchiMap UV Unwrapper python script and the layout exported in Gimp by the Save UV Face Layout script.



Left is a Gimp screen-shot of the main building textures with painted shadows.

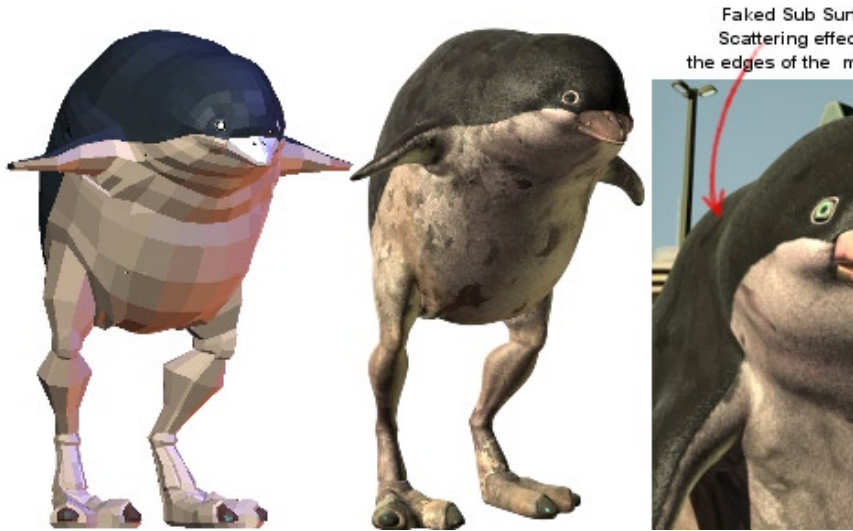
On the bottom right, the background mattes: far buildings and sky have been saved as separated layers, the buildings with alpha channel.



The penguin

The penguin has been modelled starting with a cube and extruding faces, the usual polygonal modelling (“Modelling Bongo” tutorial here). Actually I had already modelled it some time before, it was the Pengzilla (modelled just to try the new Blender’s softbody feature).

Basically the body had two different materials, the white skin and the dark skin;



several procedurals have been used to reach the final appearance. Procedurals are really useful but the more of them you use and, again, the longer are the rendering times. Image textures render faster.

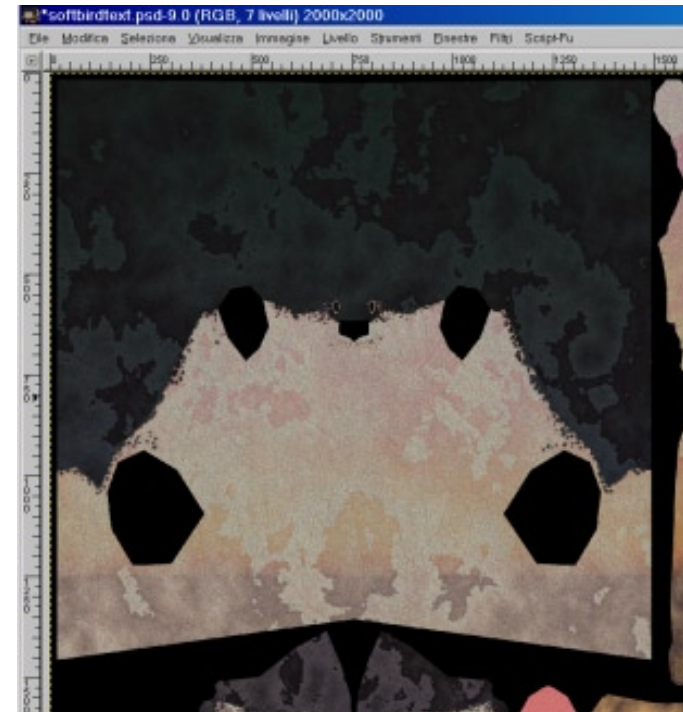
In this case, I baked the procedurals textures with the Other texture Baker script by Blenderhead z3r0_d; the script works great but sometimes mirrors in some area the baked procedurals. No problem, a bit of painting in The Gimp and everything got fixed. A damned useful script! Then, more painting to add better details.

Materials are very important too; all the penguin’s had Oren-Nayar and Blinn shaders with ramps; the ramps colours matched the colour of the sunlight and of the

ambient shadows.

A single Blend procedural mapped on the Nor Map Input (material buttons) provided a fake Sub Surface Scattering effect.

A Gimp screenshot of the baked penguin textures.



Next steps were the rigging and the skinning of the mesh (I must be honest: I usually do texturing, rigging and skinning almost at the same time. It works better to me).

With the new animation features, the rigging work flow has been tremendously improved; now it is enough to click on the screen to add bones, and by assigning the IK constraint to one of them you can quickly build the IK chain. In just a few minutes, I remade the basic armature for the penguin and then started to add constraints to automate some functions.

For example, I assigned two Action constraints to the “back” bone, targeted to the “pelvis” one: as the pelvis rotates on the left or on the right,

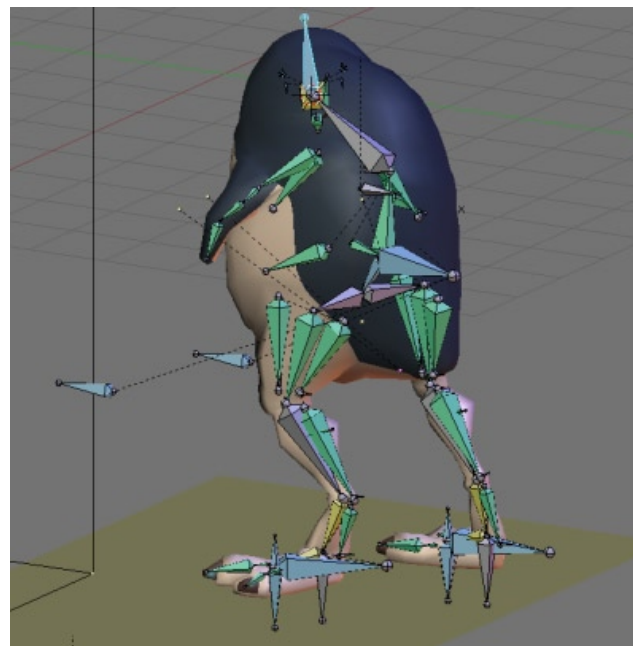
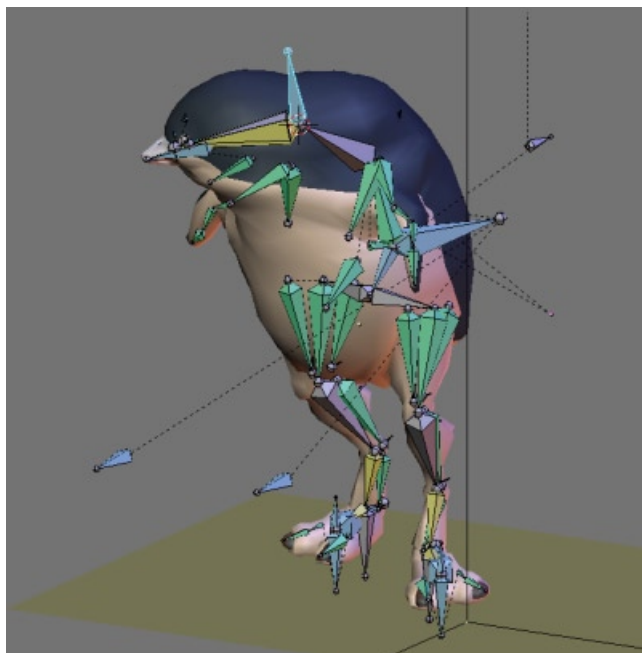
tion to balance the position.

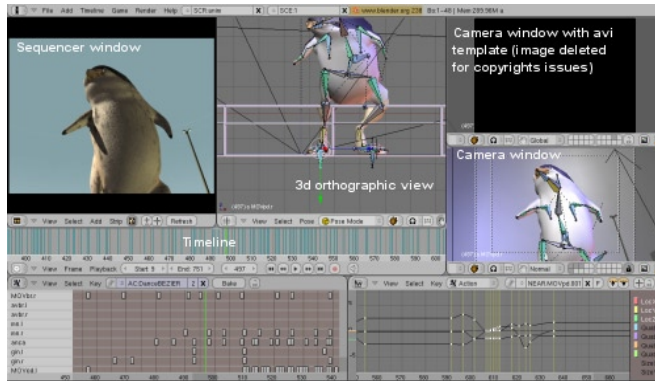
Or else, I added bones for the fat, with Locked Track constraints targeted to empties parented to root bones in the armature; the empties had a Slow Parent (in the objects window, F7) of a few frames, to create automated effects of follow through and overlapping for the fatty areas.

I added also bones with Stretch constraints in the legs areas, to simulate a minimum of muscles stretching.

The skinning of the model by creating Vertex Groups (editing window, F9) took more time; by the way, this is a too long of an issue to be covered here.

the back rotates on
the opposite direc-



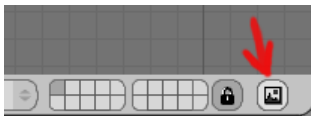


I didn't use the Non Linear Animation in this case; the whole animation has been made in a single Action. Using the original spot template as reference for the key positions, I animated with the Pose to pose technique.

In short: assign KeyFrame positions for the character, setting all the animation IPO's to constant (in the Action Editor select with A all the keys; go to the Key menu in the header and then Interpolation Mode>>Constant). This is to establish the animation times. Then switch the keys to linear and start to add in between, creating also arcs in the movements. Third, switch the keys to bezier and shift them where needed to avoid actions happening all at the same time.

In the IPO window (IPO Curve Editor) select curves, go in edit mode and modify the curves

handles, to slow or speed up starting and ending of single movements. Of course I had to rearrange the acting to conform it to the penguin shape,



quite different from the dancing robo-car. I made several tests of the animation by rendering the whole in OpenGL (hold Ctrl key and press the last icon on the header of the 3d window, pointed by the red arrow in the image).

Usually I render the different elements as separated steps and then make a composite in the Blender sequencer (Ctrl+Left Arrow to change screens, or else in the SCR: menu on the top header); this time I made all the rendering in one shot. Also the quick transition from the jumping penguin to the roasted chicken version has been made by switching in middle animation the penguin mesh with a Shift+D copy with different textures, by layer's keys assigned to both the meshes (I>>Insert Key>>Layer).

But the sequencer has been used anyway to add the horizontal bars on the top and on the bottom of the framing, for the headline at the end and for the light darkening on the sides (just to mimic the C4 spot appearance).

In Blender's sequencer I loaded also the music (made with the open source software Audacity). By selecting the "Sync" button in the sound panel I could watch my animation with sound in real time, and pressing the "MIXDOWN" button Blender exported a perfectly synchronised wave file. I then combined sound and animation in VirtualDub.

Conclusion

"New Penguoen 2.38" took almost three weeks of work in my spare time, plus 40 hours for the rendering. I rendered it (at the resolution of 800x600) with motion blur enabled at 16 (middle of render window, F10), to discover at the end that with mb the slow-

parenting effect was barely visible! So I rendered it again without mb, and except for the first and the last shots (where slowparenting weren't needed) that's the footage I used.

A lot of things can't be covered here: the detailed building of the armature and the skinning of the mesh, the atmospheric perspective effect for the 3d elements, the egg disappearing at the beginning of the animation, many materials tips to increase the reality illusion, the painting of the textures and of the background mattes and so on.

Hopefully, in the future all this (and other) will be matter for a step by step book ■

Have a good time blending!



Enrico Valenza (aka EnV) is a freelance illustrator and he lives in Italy. Besides all the traditional painting techniques he used for 20 years, he started to use cg with The Gimp and Blender almost 3 or 4 years ago. Thanks to Blender, he's hopefully moving towards animation.

Resources

The Embassy Visual Effects Inc., makers of the Citroen C4 Ad Campaign
Luxology article about the Embassy Citroen C4 Ad Campaign, C4 spot location in Vancouver. *Audacity*, *VirtualDub*, *Blender Documentation* *Blender Resources*, *scripts and plug-ins* *Modelling Bongo*, *polygonal modelling tutorial for Blender*.

Making a galvanized metal texture in Gimp

--by Gaurav Nawani

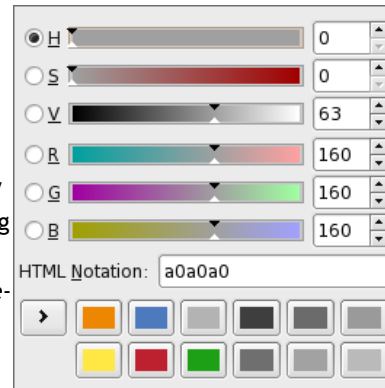
Introduction

Some times using a image base texture is preferable to the procedural texture as it provides more control over how the texture should appear in the 3d model. Hunting for a perfect looking galvanized metal texture can be a problem, 2d image manipulation programs like Gimp can be of pretty useful help in this regard. This time we are going to teach you how to make an 'galvanized metal texture' completely in Gimp in just a few simple steps.

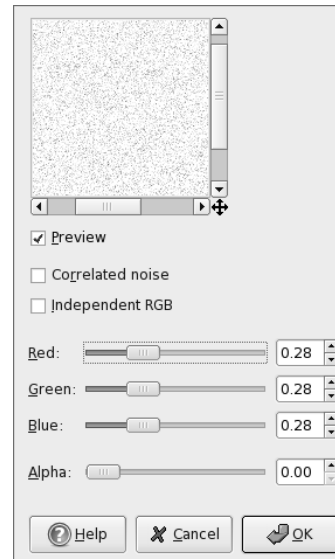
Step 1 Create a new Image in Gimp [Ctrl + N] of resolution 400x320. Create a selection of about 320x200 as shown in figure. Add a layer above the Background layer. Rename the layer to 'base-color'.



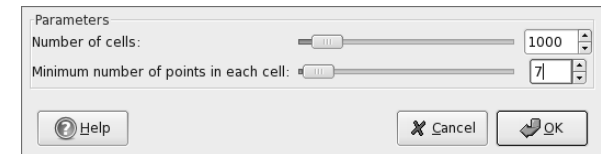
Step 2 Using the color selector dialog select the faint grey color, you can replicate the setting below in the color dialog box. Now fill the selection in 'base-color' layer with this color.



Step 3 Keep the selection and add another layer, rename it to 'noise'. Fill it with white color. Now go to 'filters | Noise | Scatter RGB'. In the pop up dialog box un-check 'Independent RGB' and move the slider to 0.28 and press OK.



Step 4 Now we will generate patterns of galvanized flakes from this noise layer, without disturbing the selection go to 'Filters | Blur | Crystalize' and change the settings as in the image. This will create the flakes that will roughly resemble the galvanized metal, and a new layer will come up the by the name of 'Crystalize'.



Step 5 The texture looks pretty flat right now. Now duplicate the noise layer and on it apply 'Filters | Blur | Gaussian blur' of 5. Rename the layer to 'noise blur' and in the 'layers stack' move upwards of 'Crystalize' layer. Change the 'layer mode' to multiply. You can also adjust the 'layer opacity' to your liking.



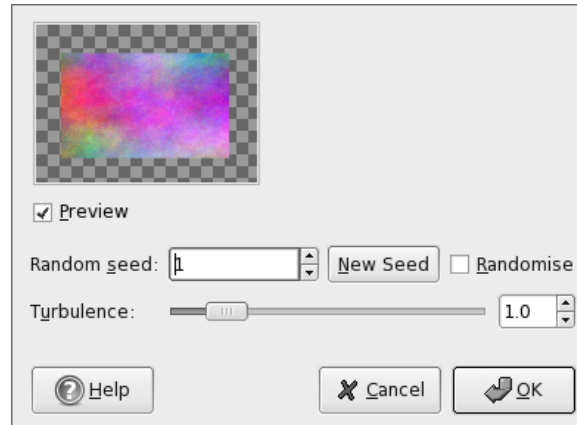
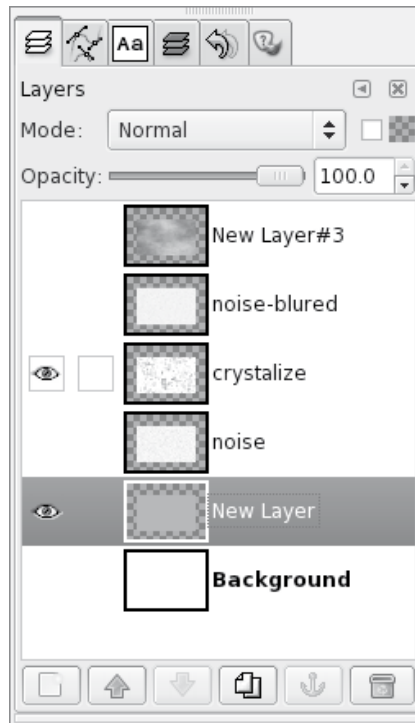
a) Before Noise



Step 6 The texture is almost done, now we need to introduce some weathering into the texture, to make it look more authentic. First I am assuming you have kept the selection alive, if not then in the layers tab right-click on the 'base layer' and do 'Alpha to selection'. Now add a layer at the top of the stack. Then go to 'Filters | Render | Noise' and generate a colorful plasma noise.

b) After Noise

The layerstack after step five.



We need only the variety of the rendered plasma noise so we will desaturate the layer. Go to 'Layers | Colors | Desaturate'. Change the 'layer mode' to 'Multiply' and adjust the 'Layer opacity'.

Your galvanized texture is now ready to be used. The Final texture on the right was created with this method and few other details added ■



Gaurav Nawani is a graphics artist at IronCode Software .

He is an open source enthusiast and loves writing articles as a freelancer.

Apart from his usual habits of photography and traveling he also dabbles with cooking often ;)

Animation Tips

-- by Sandra Gilbert

Introduction

Just like any art form, when it comes down to it, it isn't the tool being used, it is the artist and his/her knowledge/skill. Animation is no different. While a fancy program with lots of cool tools can make an animator's life easier, the animation will still fall flat if not done properly.

It has long been held in the animation industry, that there are 12 basic principles of animation. You can find them written, re-written and paraphrased all over the net. With the advent of Blender's new animation features, it is a good idea to review these principles before we all get animating.

That's right, here is another incarnation of the golden animation rules. :)

At a glance, here are the 12 Basic Principles of Animation:

1. Squash and stretch
2. Anticipation
3. Staging
4. Straight Ahead Action and Pose to Pose
5. Follow through and Overlapping Action
6. Slow In and Slow Out
7. Arcs

8. Secondary Action
9. Timing
10. Exaggeration
11. Solid Drawing (same or different as Weight)
12. Appeal

Squash and Stretch

One of the most important elements to master as an animator. It gives the illusion of weight and volume to characters, facial animation and objects such as a bouncing balls. It can also be useful while animating dialog. One thing to remember when using squash or stretch, is to always maintain the same volume. Think of it as a balloon filled with water. Squish your balloon, the water moves to different sections of the balloon, but always maintains the same volume. Tools in blender that can help in your use of Squash and Stretch are lattice deformations, Soft bodies and Shape Keys.

Anticipation

Anticipation the act of starting a movement before the movement actually starts. Generally staged as a backward movement just preceding the forward movement. It can be done broadly or subtly, depending on what effect you are going for, with broad anticipation usually used for comic effect.

Staging

Staging is the animated equivalent of photography composition. Use staging to convey your actions

clearly. Proper use of close-ups, medium and wide shots can help direct your audience to the tone, mood and action of your scene. Be sure to plan your shots and scenes out carefully so as not to confuse your audience.

Straight Ahead and Pose to Pose Animation

Straight ahead animation goes from frame to frame, allowing complete control over movement. Pose to Pose is the use of KeyFrames at set intervals, allowing the computer to fill in the missing frames. Both methods have their uses and most likely you will use a combination of both. In blender it is easy to set Pose to Pose with KeyFrames at your set intervals and then fine tune those actions on a frame by frame basis using IPO manipulation.

Follow Through and Overlapping Action

By observing the world around you will notice that objects in motion never stop all at once. In animation this is called Follow Through and Overlapping Actions. Most obvious examples hair flowing and settling into place once the character has stopped. An animal with loose skin, different parts are all moving, but at different speeds and stop at different points.

Slow-Out and Slow-In

Slow-out and slow-in will soften an action and render it more life like. The basic concept for setting them up is as follows. You bunch up the KeyFrames toward the beginning and end of each action with only one or two in the middle of the action. More KeyFrames creating slower actions, fewer KeyFrames creating faster actions.

Arcs

Remember your math teacher saying you'll use this someday, here it is. Most all movements are on an Arc. A soft semi circular motion that gives a more natural feel to the movements. Test it yourself, raise your arm and move it around, there is your arc. Look for arcs in movements all around you.

Secondary Action

Secondary action provides the extra little details that make a performance more believable. Adding arm movements to a walk cycle, head bobs, and dialog all are examples of secondary action.

Timing

Timing can make or break an animation. It is one of those things that is best learnt through trial, error and experience. A stopwatch can greatly increase your grasp of timing. Go into a room, shut the door and act out your scenes, using a stopwatch to record when the actions start and stop. Don't forget to overlap actions from different characters and objects.

Grab a friend and play out your scenes that involve more than one character

Exaggeration

In computer animation it is very easy for your movements to look stiff and staged, minor exaggeration of features and movements can soften your movements and make them more fluid. But be careful with how much you exaggerate, unless you are going for a very comic look.

Solid Drawing

Solid drawing covers the basic principles of giving form, weight, and volume solidity to your objects. This actually applies more to traditional hand drawn cell animation, but should be kept in mind even for CG animators. Make sure that your characters and objects have the look and feel of a solid object. Check from all angles to ensure that you haven't forgotten something.

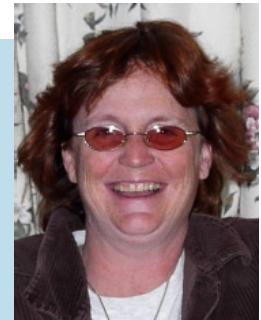
Appeal

Often we get so caught up in all the mechanics of animation that we forget all about appeal. The whole reason for making the animation was to tell a story, is your character believable, will it connect with the audience? These are questions you need to keep in mind when creating your characters. Make sure they have personalities, even if they are evil ones.

Okay now we have covered the very basic of the

basic principles. It gives you the ideas and concepts to improve your animations, but like all things in life, this is a subject that requires further study and a lot of practise ■

Hi, my name is **Sandra Gilbert**, (aka dreamsgate). I have been using blender for a little over five years. I currently live in Nampa, Idaho, USA and work as a graphic designer for a small print shop.



I am married and have 2 children, which leaves not near enough time for feeding my growing blender obsession. Yet somehow I always manage to find the time to explore new features, keep up on the latest news and start new blender projects. Some of them I actually even manage to finish.

Project Orange Interview



by - Sandra Gilbert

Have you ever sat back and watched a project or idea literally explode across the scene. Now for the blender community this is not an unusual occurrence. We have a long history of avidly supporting any community project, idea or fund raising campaign that comes along. But even with our history of support, the popularity and support gathered around Project Orange has been gaining an historic momentum. News of which has appeared not only on community websites, but has leaked

over into Open Source news sites and even been featured in issue #68 of 3D World where the Project Orange website had been chosen as Website of the Month.

Blender, having been an in house production tool, once had the immediate feedback and interaction of both developing and using the software on a daily basis. With a desire to recapture that unique synergy of development and a very real desire to create a professional quality Open Source animation, Project Orange

was born. The Blender Foundation teamed up with Montevideo (Netherlands Media Art Institute) to produce and distribute the resulting animation. With Montevideo handling the tax/legal and hiring issues and the Blender Foundation taking charge of the creative process, a new focus was about to hit the blender community and take hold of our imaginations.

From a basic idea, a full fledge studio soon appeared. Two apartments were rented and equipment was set up. Jan Morgenstern and his studio WaveMage

agreed to sponsor the movie by providing all music and the full sound effect editing for the project. Next Prof. Mark Alan Matties, Department of Computer Science at Bowie State University and Director of the BSU Xseed xseed.bowie-state.edu, offered the BSU Xseed as a rendering farm for the entire period of the Orange Open Movie project. The offer of a full render farm allowed for more creative scenes using longer render times, while still maintaining the set production schedule.

And yet as all the organisational stuff is going on, the real story revolves around a team of brilliant artists, who voluntarily locked themselves into a studio in Amsterdam to work 16 hour days. Bouncing ideas off each other and helping solve road blocks and problems, the Orange team has been hard at work creating a style that will forever be all their own.

When I decided to write this article and had gotten permission to use information from the website, I thought it would be a simple thing to summarise all the information they have provided on their website. That didn't prove to be the case. They have posted such a wealth of information that it can't simply

be condensed into a short article. Having followed the Project from the beginning, I hadn't realised how much they had shared on their website until I went to download it all and try and put it into perspective. Months of work has been documented and posted for our education. Their work and dedication to such a huge undertaking is nothing short of amazing.

They have pushed the envelope of blender capabilities, finding its strengths and weaknesses. In fact their work and close communication with the coders has pushed the development of blender features into hyper-overdrive, bringing the community a new wealth of tools for our own enjoyment and projects. Such as the addition of a better particle system, making realistic hair far easier to achieve. And Shape keys have added a much easier way to deal with facial animation. The list of improvements and new features added to blender is already overwhelming to say the least.

Throughout the last six months their work has fired the imaginations of an entire community. Blender users the world over are firing up blender and attempting for the first time their own animations, inspired by Project Orange and the new ease of use for animation features.

We have watched the first concept sketches turning step by step into a movie. Feature tests and progress reports tease and delight us with the endless possibilities such projects bring. Keeping with the Open Source/ Creative Commons ideal they have freely shared each step of the process, educating the community on new techniques as they discover them themselves.

As the weeks have passed, they have tested and refined each stage of the process. Now having passed the half way point in the project, characters are built, scenes set up and basic timing in place. With each further week that passes, the scenes will be further refined and moulded into shape. All the while Jan Morgenstern has been working closely with the team to create the music and mood for the movie.

As they are approaching the final stages of their production they are pushing even harder to get everything organised and finished up. Voice recording

for the characters has been completed with well known Dutch actors Tygo Germandt as Proog and Cas Jansen as Emo. They videoed the recording session which now gives them great reference material for animating the expressions and movements of Proog and Emo. From this point on the team will be continuing the fine tuning of every little detail and polishing it all up.

Knowing all too well how busy the team members are at this point, I bravely or foolishly (depending on your view) decided to try and get an interview with the Project Orange team members, hoping at best that at least one would have the time to answer a few questions. To my great delight, five of them found the time to answer. So



without further delay, here they are.

1-Aside from the official goals of the project, what do you personally hope

to accomplish or take away from this experience?

Lee J. Cooks (LohnC): Ah this is a question with a thousand answers hehe. I think mainly working with a team on a project has been something very new to all in the team, and being halfway through the project we have been really working out our specialities, and how to work together well. I hope and believe we will have learnt a lot about forming an idea of many, and executing it. We may be like brothers by the end.

Toni Alatalo (Antont): Learn about all aspects of movie making. And also get professional in Blender and related software development. It has proved to be fun too and a great group of people to get to know to, which is very rewarding considering many aspects of life.

Andy Goralczyk (@ndy): I would really love to make a *good* movie that not only I enjoy watching, but also a movie that's appealing to many other people. And I really hope that this is really not the average boring CG short, we hope to treat it as art, as a real movie. Besides, I just want to have fun in the process and learn... that's

all!

Bastian Salmela (basse): Hmm..., personal goals.. I would say.. to learn how to actually make a movie, like the big boys do, and also what this weird blender program is about.

Bassam Kurdali (slikdigit): Spending 6 months making a short animated movie is my biggest personal motivation; it's what I would most want to do with my time. Beyond that goal, I'd say learning plays a big part in my motivation: becoming a better animator, learning from my teammates' substantial abilities, learning more about film making. I also wouldn't be displeased if project orange enabled me to more easily do similar projects in the future.



2- As busy as you are and as fast as new features are being coded, do you find it hard to keep up and master the new tools?

Lee J. Cooks (LohnC): Wow, hasn't it been amazing the speed of everything lately. Many times we will form a great idea among the team that could be great for the community and in our work flow for the film, after we have told Ton that night he has hacked away and the feature is there. Since we are needing to use it sometimes in that week for the film, and also wanting to get the feature optimised as soon as possible, many late nights have come from trying to learn all the great new things in Blender. I can admit I try to master a new feature added, but most times you just have to learn as much as possible of how it works before your using it, and of course the next night a new one is there!

Toni Alatalo (Antont): Every day is learning.. for me, as a kind of a developer of the tools, the perspective is of course different .. have a lot of time for learning.

Andy Goralczyk (@ndy): We're basically working with blender 15 hours a day, every day, every week... so we're learning new things as they come up. Our luxury is that most of the new tools that drop in are specifically coded for our needs. It takes of course some time to get used to everything, but so far it

wasn't that hard. It's great to see the software evolving as you're using it

Bastian Salmela (basse): YES!!! heh, I think I am the only one at the studio complaining that there is too much new things popping up in CVS everyday. I just don't have time to learn all of them. but of course, these are the features we asked, features we need, so I try my best to keep up.

Bassam Kurdali (slikdigit): Naaah! OK, well, sometimes At first I would play with every new feature as it came out, but now the schedule is a little rough for that. It's easy for me to stay on top of the animation features, since I'm the one who asked for half of them I let Andy worry about the materials and rendering stuff- I haven't even done a test render with the new materials layering system, but I promise myself I'll spend some time playing with it after the project is over.

3- So far what has been your biggest learning experience on this project?

Lee J. Cooks (LohnC): I can say there are more things then all the fingers in the world could count that I have learnt so far from this wonderful and hectic experience. Probably the biggest learning experience was the production work flow. The idea (as happens in feature films too) can change slightly, and improve every day more and more,

also some models or character rigs are always being worked on and replaced (sometimes due to a new feature). The flexibility to work with Bassam on keeping everything consistent and keeping up with the ever changing film along with an ever changing program has been by far the biggest.

Toni Alatalo (Antont): Dunno if it the greatest, but a nice basic lesson is that in animation programming it is not needed to control frame-by-frame like in game logic, but things can be done nicely by creating IPO curves etc.

Andy Goralczyk (@ndy): The main driving force behind the learning process is that we all have different fields of experience. From my fellow team members I learnt so much more about animation,

cinematography, lighting, modelling and also the technical aspects of rendering than in any other project before.

Bastian Salmela (basse): The beer in Amsterdam is much cheaper than in Finland, and water in Amstel doesn't kill you, just remember to shower afterwards.

Bassam Kurdali (slikdigit): This is my disobedience to authority workshop I can't say I've made huge strides so far, but hopefully I'm getting a little better. I'm also learning a lot about making a short movie of this type, especially with the early problems we had getting the script /story stabilised. I don't think I'd start production again on a project



without first having a script completely finished. The planning for the project is also quite interesting. It's amazing how much structure and speed it gives us, compared to my usual "planning without planning" approach.

4- Project Orange has become the focus of considerable attention; how do you feel about so many pairs of eyes watching everything you do?

Lee J. Cooks (LohnC): Haha well it is very weird when somebody like the Ambassador of European Media comes to have a gander at what we are doing, sometimes I must admit being in the studio each day for 3 months now it is hard to believe we would have such attention, each day has its own little challenges and we usually have to concentrate on these, but sometimes you get a little shock here and there to realise we are being followed .

Toni Alatalo (Antont): They don't see me, I am well hidden and unknown. I guess we mostly just keep forgetting that.

Andy Goralczyk (@ndy): On some ways it's pretty helpful, we're getting a lot of feedback on our project, and not only on our website! A lot of people from our host, the Netherlands media

art institute are constantly helping and supporting us, too. On the other hand there's also a lot of noise (mainly from the Internet) being thrown at us. At some point you just have to push everything away and concentrate on your job.

I'd also like to add, that contrary to many overly unrealistic expectations of a certain group of people, we really are not Pixar or DreamWorks or SPI! We're in no way even attempting that. To be realistic we neither have the talent, the experience or magical substance. But we are studio orange!

Bastian Salmela (basse): It's like being in Big Brother. Except that you are stuck with us, there is no way to vote anyone out of here.. hmm, but yea.. well I like sitting back and to watch peoples reactions. Sometimes they get very excited, sometimes angry.. depending if we are doing something right, or wrong.. And they always have lot of opinions. That's always fun.

Bassam Kurdali (slikdigit): It's no problem really. We all just behave like we normally do. Andy issues dire threats from his desk, Lee contentedly munches on brains, Basse paints inscrutable paintings with increasingly esoteric foodstuffs (he's using Swedish herring

and milk now) while Toni and I indulge in the pleasures of Amsterdam. In all seriousness, we're mostly very focused on the work we do, but the online comments provide us with a nice break from time to time. Most people in the community seem enthusiastic about the project, and have contributed freely their insights and ideas.

5- As hard as you are working to meet project deadlines; has this experience turned out to be more or less "fun" than you originally expected?

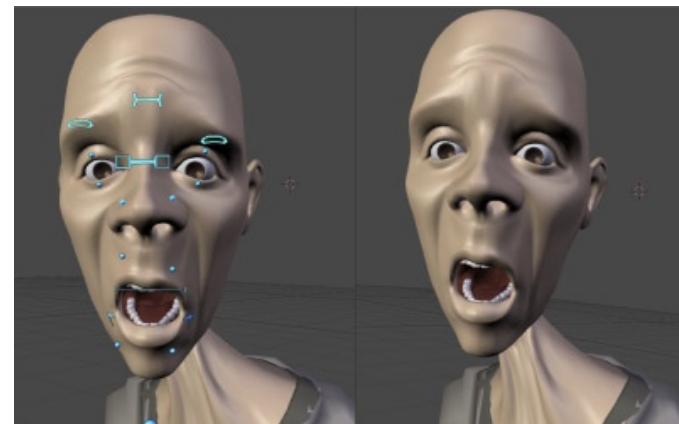
Lee J. Cooks (LohnC): Oh, this is another hard one. It's probably safe to say that if I thought one thing before the project started, when it did start all my expectations were flipped upside down. It really has been a lot more fun then I could have hoped for, but in such an opposite way then I could imagine before I started the project. The challenges, hard work, sharing ideas and ogling over others screens after they finish a

shot with wonderful results have been amazingly fun, like having a very long game of Tennis, It's exhaustingly fun. You learn to blend longer without as many rushes of excitement, but when they come around they are a lot more satisfying.

Toni Alatalo (Antont): hm .. perhaps more, or well I don't remember any more what I expected

Andy Goralczyk (@ndy): It's actually more fun than I expected! We all knew from the beginning that it would be hard work, but at the end it will definitely be worth all the effort.

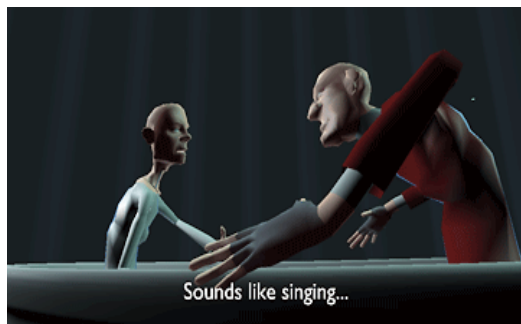
Bastian Salmela (basse): It's still fun. Sometimes it gets to you, when



you are fighting with something that just doesn't work and time is running out. But in the end, they always turn out nicely. Deadlines are very important to have, they don't only make you work hard, but they also make you to actually finish things and move on to the next thing.. just because you have to.

Bassam Kurdali (slikdigit): It's pretty close to my expectations in terms of fun..

6- What aspect of this project have you enjoyed the most?



Lee J. Cooks (LohnC): I'll try keep this one short and sweet ... *think for a few hours*. OK, most enjoyable thing about the project would have to be having a technical director! If anybody can break a computer It's me, and now there is somebody around who knows how to

fix it!

Toni Alatalo (Antont): Perhaps being allowed to concentrate on interesting things, like animation scripting .. but also being a part in the whole process.

Andy Goralczyk (@ndy): I think I'm enjoying mainly the fact that I'm able to work with people who share the same interests, working on a common task, in the same place! and I also like the opportunity of really going insane with style, animation, lighting, everything!

Bastian Salmela (basse): Maybe the possibility to actually work full time on just designing, drawing, modelling, animating.. instead of what I've used to at home, having couple of hours to blend after midnight.. oh, and all the cheap beer of course.

Bassam Kurdali (slikdigit): Can't single anything out really.. Animating has been big fun the last few days, making some wacky stuff, some of the facial rigging has been pretty fun too.

7- Any tantalising tidbits about the movie that you would like to share with us?

Lee J. Cooks (LohnC): Well, I'm still pushing Bassam for it, but I'm debating to have the huge anime style sword fight scene, I'll inform everyone if there is any progress

Toni Alatalo (Antont): The shoes are great!

Andy Goralczyk (@ndy): Dutch cheese is gooooooooooooooooood!

Bastian Salmela (basse): I have been working with these little things that nobody knows what they are all about, and I heard somebody saying: "Oof, what are they?"

Bassam Kurdali (slikdigit): Well we've gotten some big name (in the Netherlands) stars to do the voices for our characters I'm not sure I'm allowed to reveal who it is yet, and it's likely only to excite Dutch audiences, but it's good news for us, and might help the movie get taken more seriously by non members of the

blender community, for instance in film festivals, etc ■

Project Orange still has some time to go before they wrap it all up and return home. For further information and updates on their progress, stop by the Project website. <http://orange.blender.org/>

--Sandra Gilbert



Stop Staring

Learning facial animation have always been mysterious for those unaware of the tools and techniques. It requires one to understand the basics of facial features their movements during speech, and also to understand the efficient modeling techniques, only then can this arduous task be fun.

The book 'Stop Staring' provides the user with a very tidy approach towards learning facial animations. The book itself is divided into four parts comprising of 13 well thought out chapters.

In part1 you start by learning the basics of mouth movements during speech and their lip sync equivalent in the 3d model, movement of eyes, brows, and facial land marking that supports the facial movements are also covered.

Part2 deals with introduction to Visimes. It also covers topology of a mouth 3d-mesh and finally ends with construction of mouth keys.

Part3 deals with basics of evoking emotions through movement of eyes and the use of eyebrows in conjunction. Later parts also deal with the topology of the eye and eyebrows in the 3d-mesh.

Part 4 puts into practise the knowledge learnt in the previous three parts. It also details techniques like skeletal setup, weighting and rigging with practical explanation of the animations in the accompanying CD.

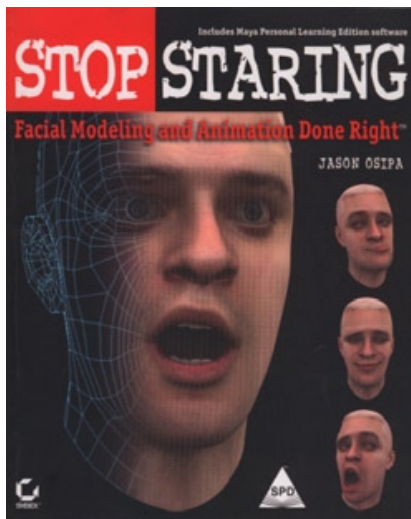
The place where this book shines is in thorough explanation of all the basics regarding the speech cycle of the human mouth and then providing a very practical way of achieving the desired results in CG, for example, most of us newbies

will think that we need to make the mouth movements reflect the actual human mouth movements but doing that is down right cumbersome whereas the author explains a logical and organised way of achieving that by providing the knowledge behind the appearance of expressions in mouth with the spoken word.

All in all the book goes through every aspect of facial animation, from the basics of human facial movements to defining and how to achieve the modeling or topology for each part of the face that plays part in bringing out emotions. By the end of this book you can expect yourself to be well versed in modeling an animatable face model, will be able to rig it and have the good knowledge of facial animation.

This book is a must read for every wannabe animator out there ■

- *blenderart*



Name: Stop Staring, Facial modeling and animation done right.

Author: Jason Osipa

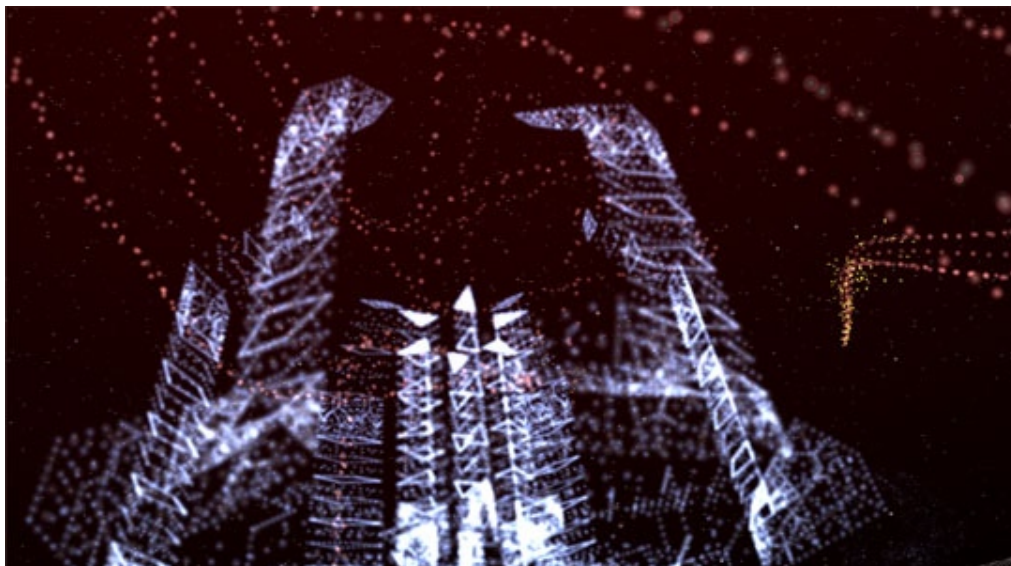
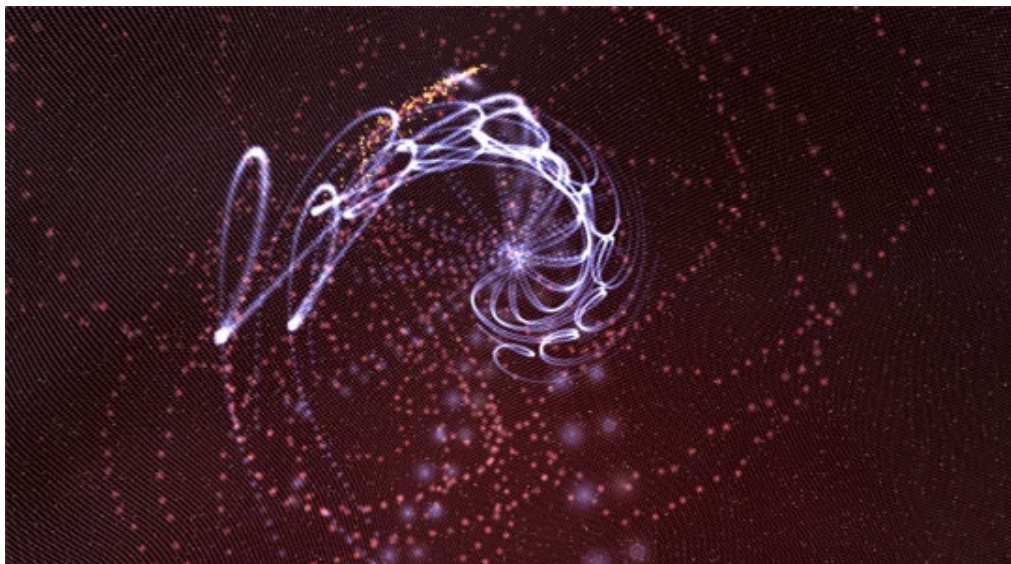
Paperback: 334 pages

Publisher: Sybex (Dist SPD India) & CD Rom

Edition: September 2003

Language: English

ISBN 81-7366-815-9



Esign
Chris Larkee (knellotron)
Award: Best Animation Artwork



*Jake Rocks
Thorsten Schlueter
Nomination: Best Animation Artwork*



Alchemy Trailer
by Jason Pierce
Nomination: Best Character Animation
"Copyright 2005 Jason Pierce. Used by permission"



treefrog.nature
by Jason Pierce
Nomination: Best Animation Artwork
"Copyright 2005 Jason Pierce. Used by permission"

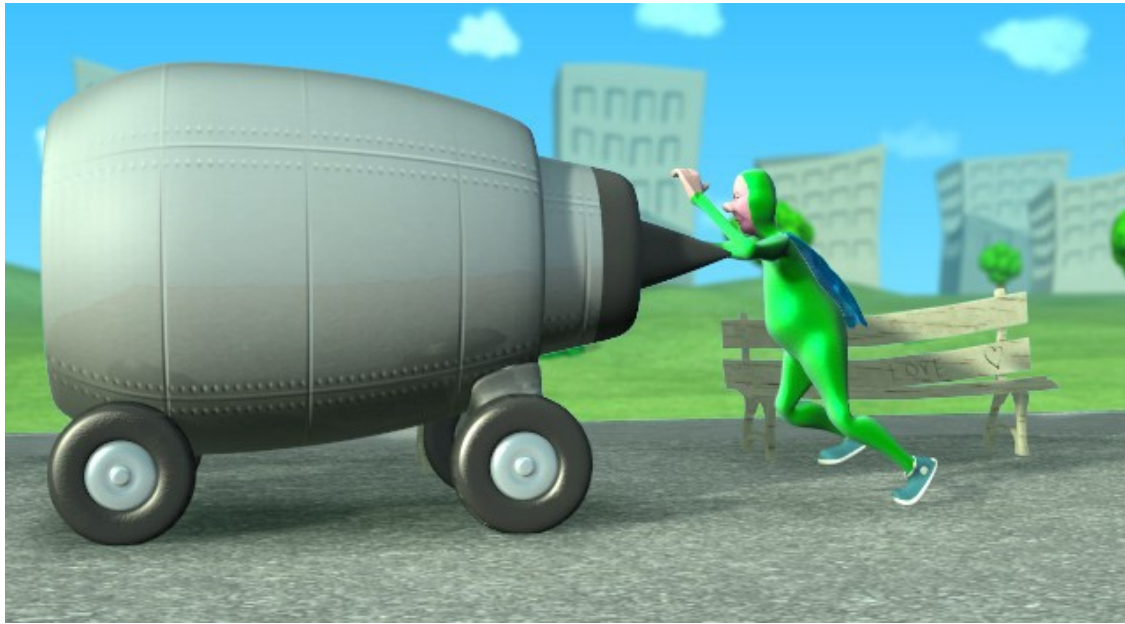


Laws of Motion

Robert Tiess (Robertt)

Nomination: Best Character Animation

link: http://www.blender.org/cms/Animation_Festival_200.718.0.html



Learning to Fly
by Grzegorz Rakoczy
Nomination: Best Character Animation

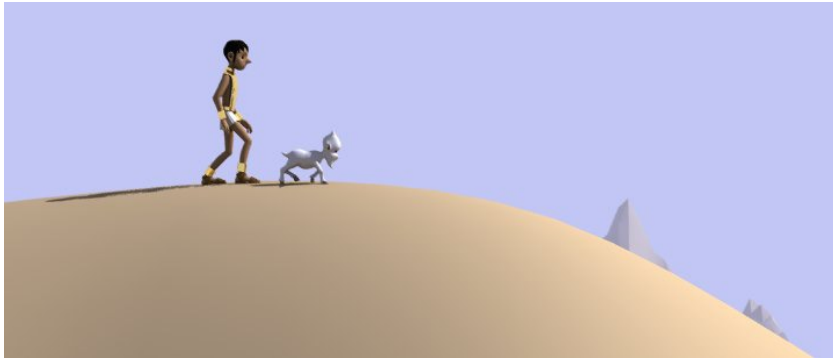


New Penguoen
Enrico Valenza (Env)
Award: Best Animation Original Idea or Story

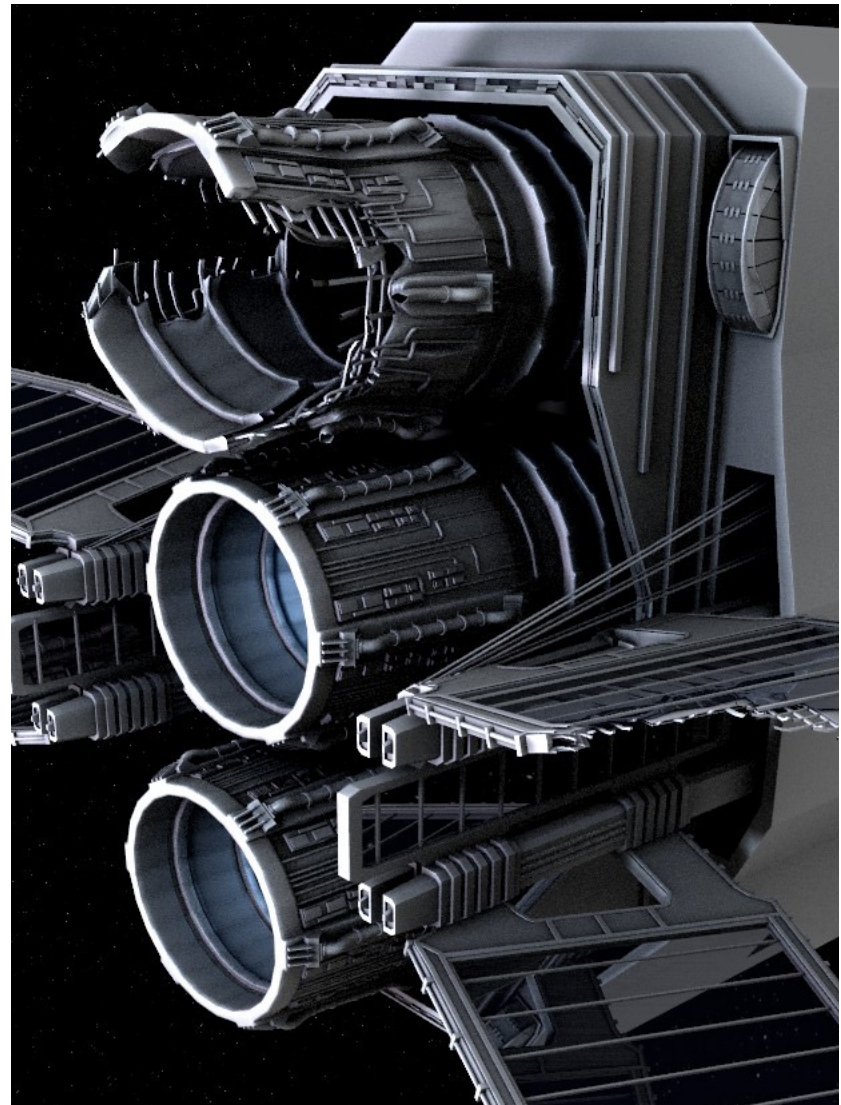
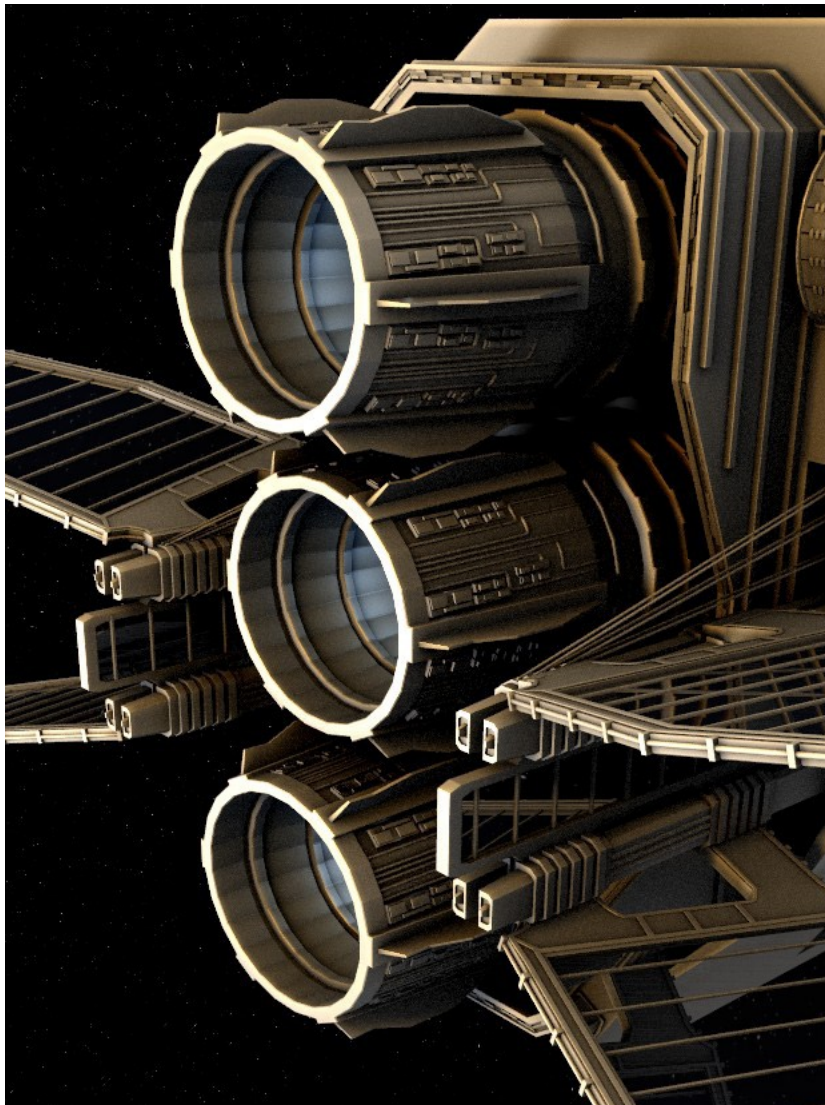




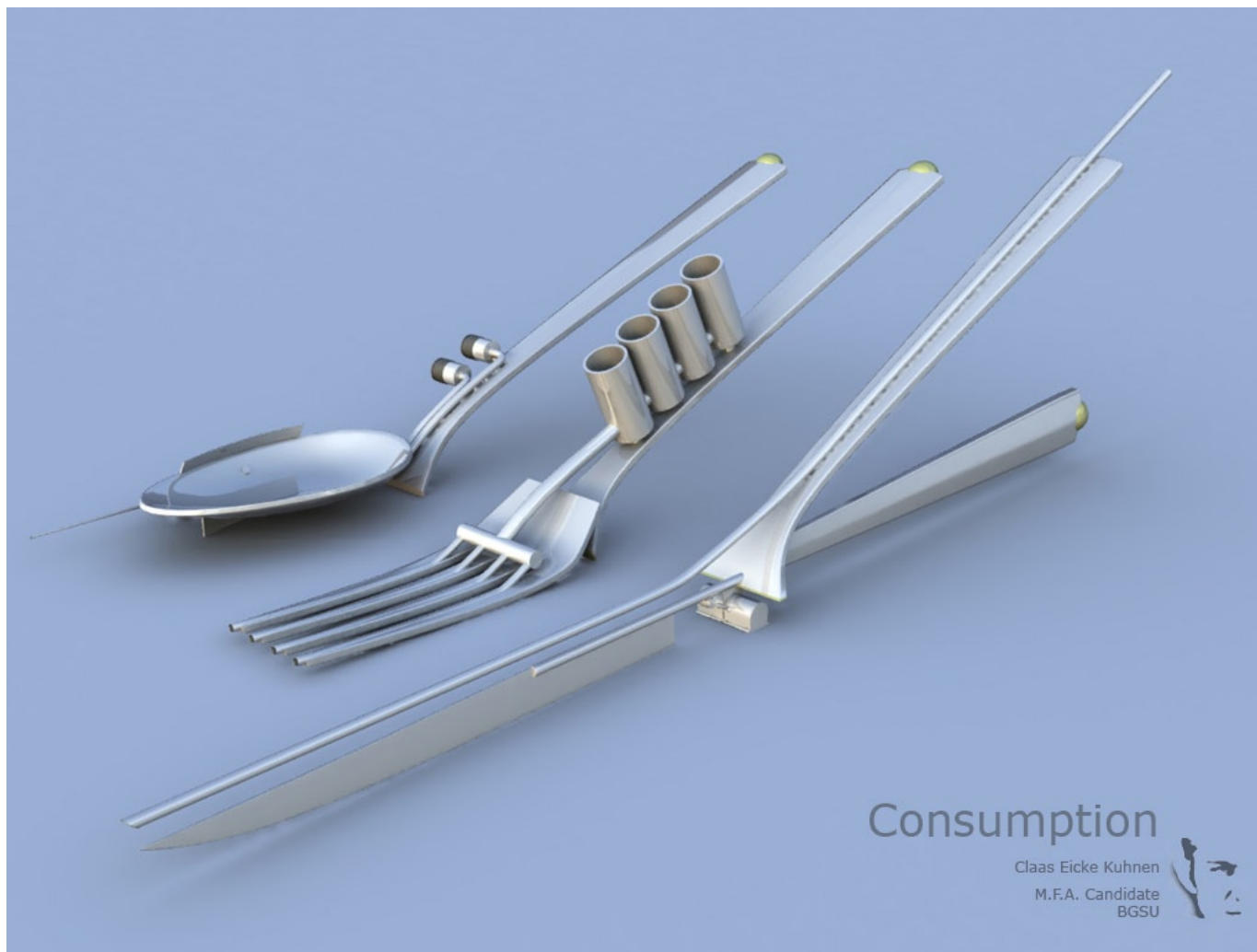
Plumiferos
by Studio Manos Digitale
Nomination Best Animation, original idea or story



The Goat, the Boy and the Sun
by Martin White
Nomination: Best Animation Artwork



The Exodus Intro Animation
Ecks

**Goofup :P**

This artwork was sent by Cekuhnen for the Issue1 of blenderart but we were not able to publish it due to some error in communication among us. Apologies to Cekuhnen.

Also congratulations to Cekuhnen on being awarded best piece in a show at the SOFA exhibition for this image.

Consumption
Cekuhnen

ISSUE 3

Available in March 2006

Theme: Rendering

HIGHLIGHTS

Blenders rendering engine explained

Various lamp types and tricks

And more...

DISCLAIMER

blenderart.org does not takes any responsibility both expressed or implied for the material and its nature, or accuracy of the information which is published in this PDF magazine. All the materials presented in this PDF magazine have been produced with the expressed permission of their respective authors/owners . blenderart.org and the contributors disclaim all warranties, expressed or implied, including, but not limited to, implied warranties of merchantability or fitness for a particular purpose. All images and materials present in this document are printed/re-printed with expressed permission from the authors/owners .

This PDF magazine is archived and available from the blenderart.org website. The blenderart magazine is made available under Creative Commons 'Attribution-NoDerivs2.5' license.

The CC license is available at <http://creativecommons.org/licenses/by-nd/2.5/legalcode>